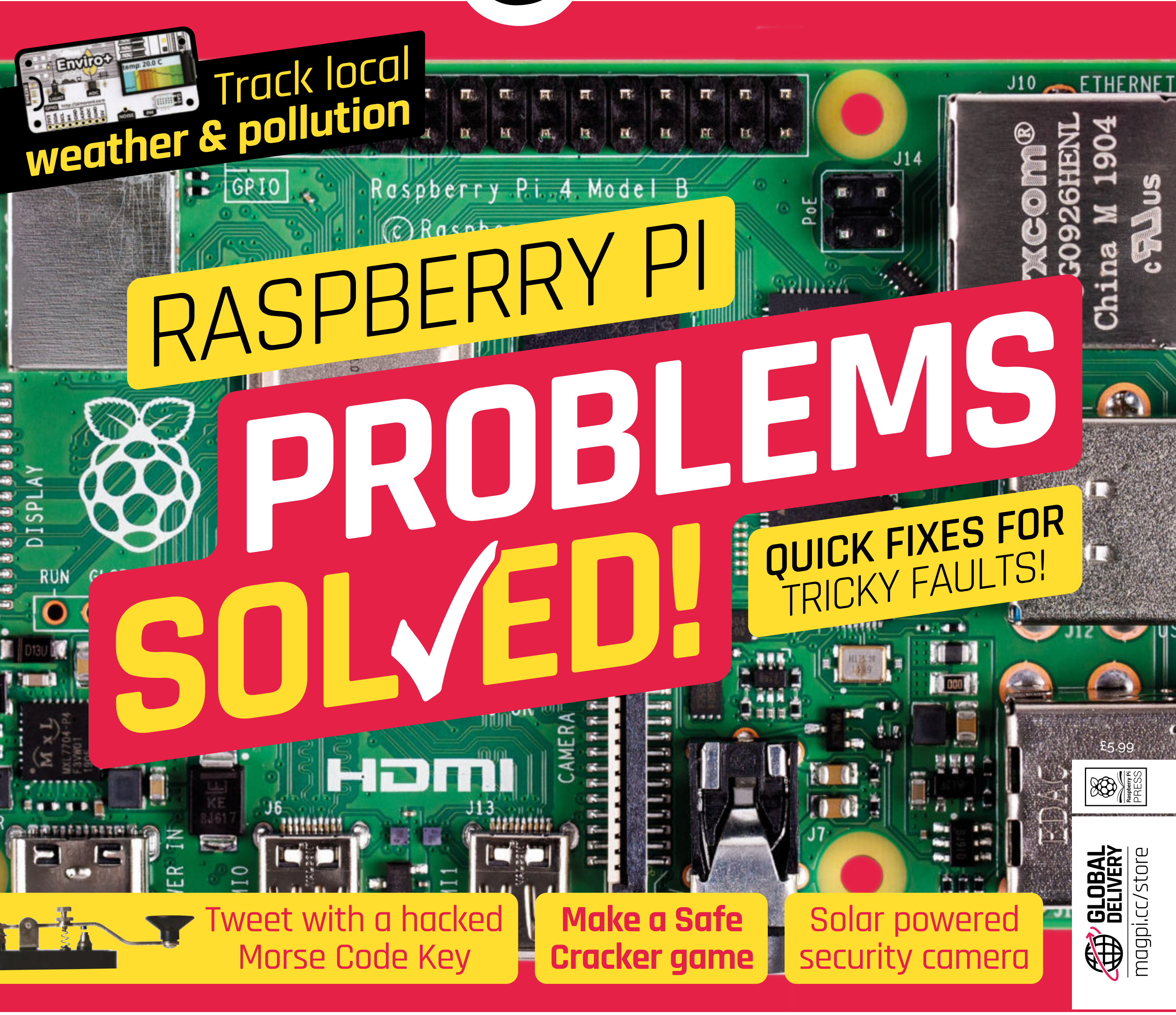
The MagPi

Issue 92 | April 2020 | magpi.cc

The official Raspberry Pi magazine



Track local weather & pollution

RASPBERRY PI PROBLEMS SOLVED!

QUICK FIXES FOR TRICKY FAULTS!

Tweet with a hacked Morse Code Key

Make a Safe Cracker game

Solar powered security camera

£5.99

43 PAGES OF PROJECTS & TUTORIALS

Augmented Reality
Digital Omnipolar
Switches
Connectors
Automation
Crystals
Passives
Circuit Protection
IO-Link
Interconnect
Logic
Electromechanical
Semiconductors
Hyperfast
Power
IoT
RFID
Keyfob
MCUs
Solenoids
Axis Tilt
Embedded Computers
SPI Interface
Capacitive Touch
Proximity Sensor
Shunt Sense
5G Mobile
Virtual Reality
RF Antennas
Microservice Architecture

WELCOME
to *The MagPi* 92

If you have a problem, if nobody else can help, and – crucially – if it's Raspberry Pi-related, maybe you should buy *The MagPi* magazine. OK, so we're not quite the A-Team, but we are experts at troubleshooting computers. Rob has put together an incredible Raspberry Pi Problems Solved feature (page 28). From fixing pesky boot problems to advanced system support, there's advice for every maker.

We've got some top tutorials too. PJ has now added facial recognition to his magic mirror (page 50). He's also found time to hack a Morse code key to send messages to Twitter (page 42). I've been continuing to delve into Raspberry Pi Shake earthquake data (page 46). Rosie and Phil have been looking at weather and pollution tracking (page 60).

And that's before we get to the community projects: from a CNC machine hacked to become an Etch-a-Sketch (page 14), to a useful sourdough incubator (page 20), to an amazing clock that uses Instagram images for its digits (page 22).

We solve Raspberry Pi-related problems, and the incredible community never lets us down. So hunker down, ignore the outside world for a while, and enjoy a cracking edition of *The MagPi*.

See you next month!

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy is editor of *The MagPi* and an expert in her chosen fields of retro computing, imaginary dogs, and pasta making. Self-isolating since the 1980s.

magpi.cc

WIN! ARGON ONE THERMAL CASES
The MagPi
The official Raspberry Pi magazine
Track local weather & pollution
RASPBERRY PI PROBLEMS SOLVED!
QUICK FIXES FOR TRICKY FAULTS!
Tweet with a hacked Morse Code Key
Make a Safe Cracker game
Solar powered security camera
43 PAGES OF PROJECTS & TUTORIALS

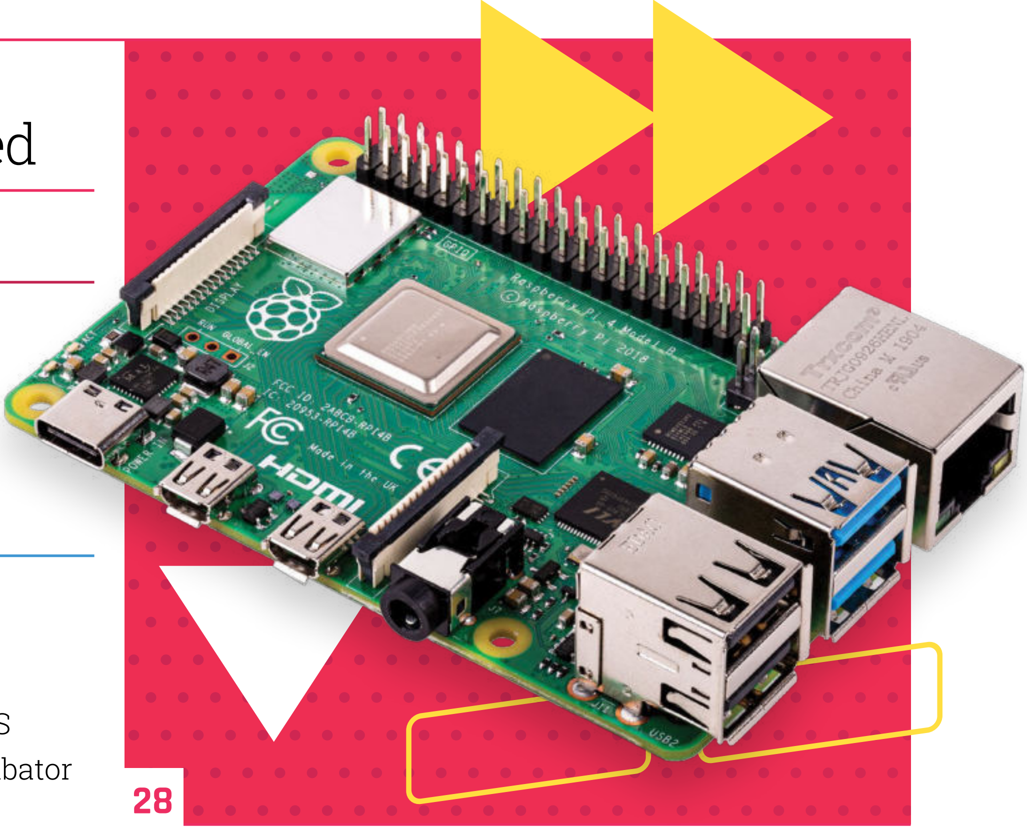
Contents

> Issue 92 > April 2020

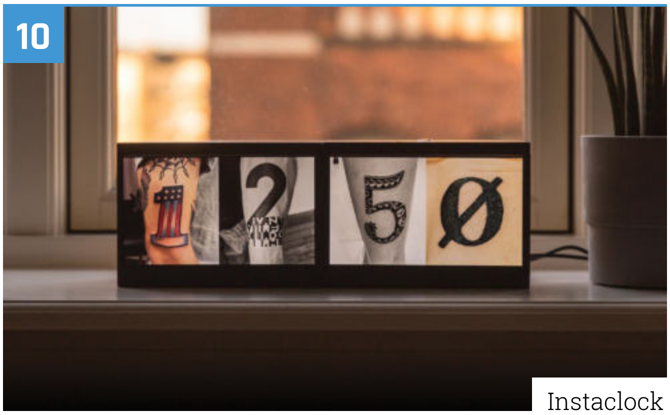
Cover Feature

Regulars

Project Showcases



28



10

Instaclock

18

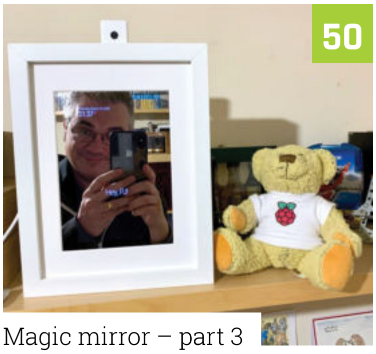
Flying Saucer Clock & NAS

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

42

Tweet with Morse code

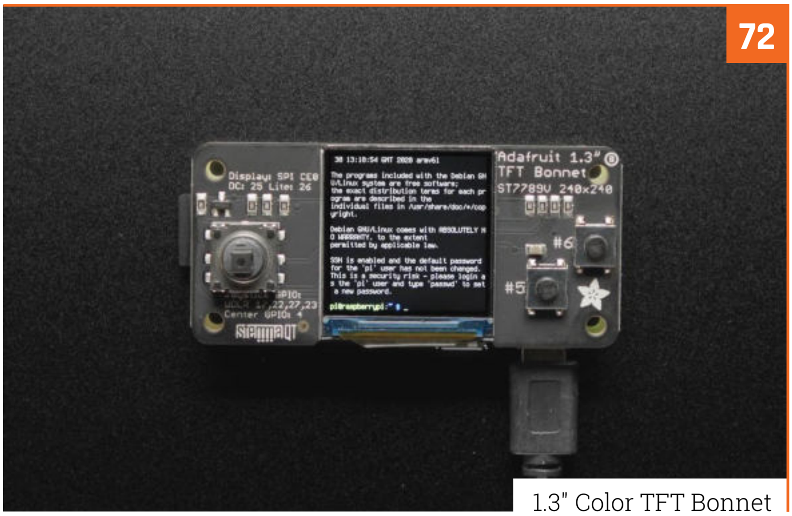
50



Magic mirror – part 3

Tutorials

72



1.3" Color TFT Bonnet

The Big Feature

60

Track weather & pollution

Reviews

82

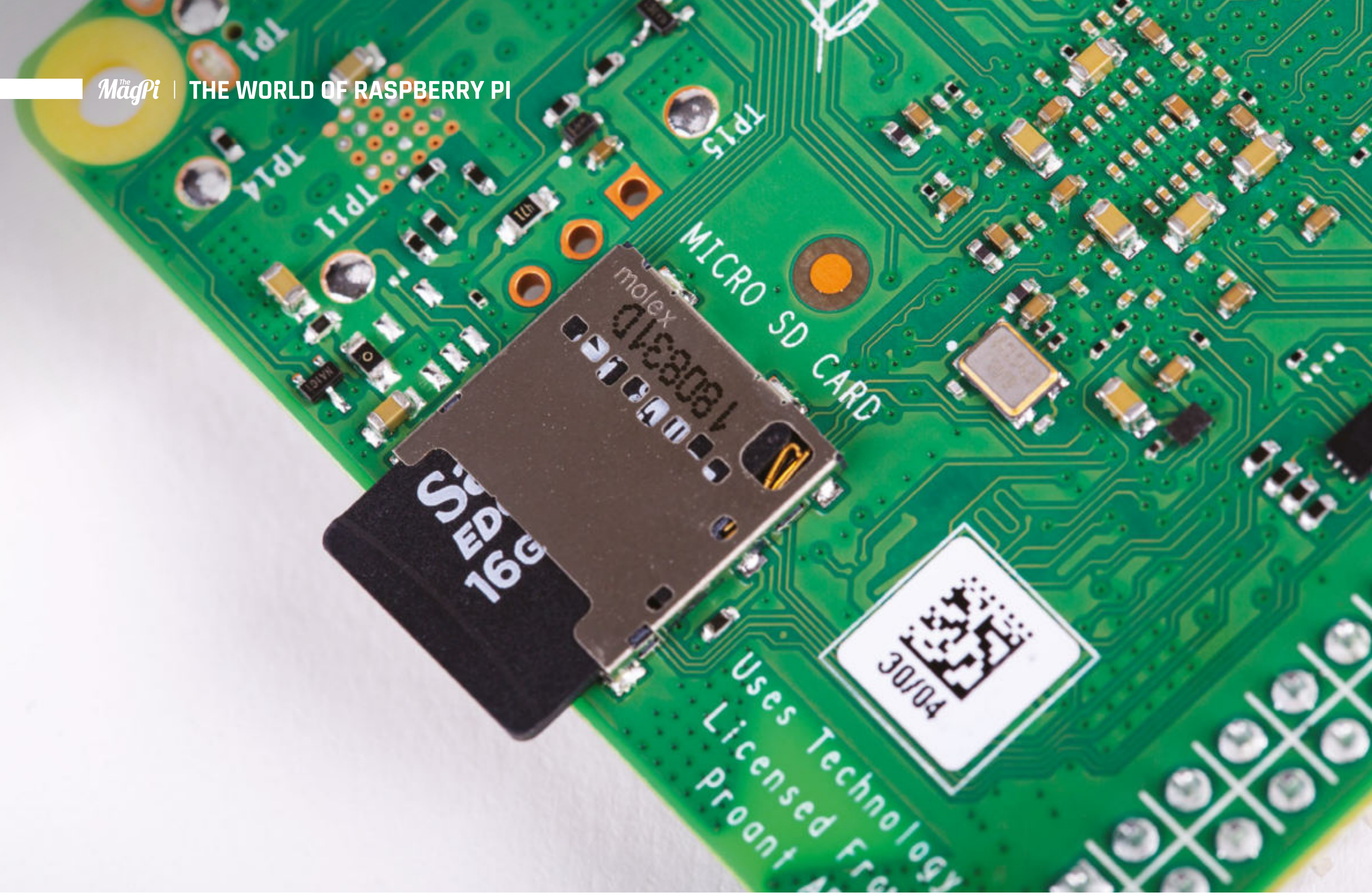
Dr Wednaud Ronelus interview

Community

WIN

ARGON ONE
COOLING CASES

95



Raspberry Pi Imager released

New official tool makes it easy to write an OS image to your microSD card. By **Rosie Hattersley**

▲ Until now, beginners had been advised to buy a kit with Raspberry Pi software pre-installed on a card

Raspberry Pi has launched an OS imaging tool called Raspberry Pi Imager. It makes it simpler to reuse an existing microSD or SD card with Raspberry Pi.

The new tool enables Raspberry Pi users to install the free Raspbian OS (available in three versions) along with a range of other operating systems (**magpi.cc/introducingimager**).

Gordon Hollingworth, Director of Software Engineering at Raspberry Pi, explains: "One of the most important aspects of the Raspberry Pi experience is trying to make it as easy as possible to get started."

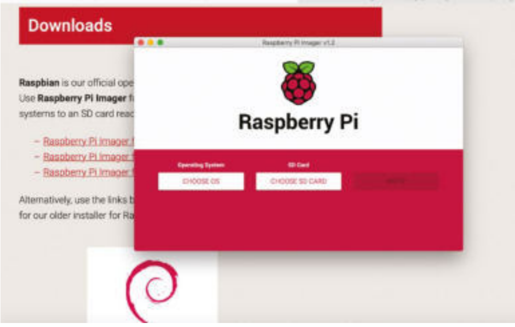
Despite options such as NOOBS and Etcher, "when it comes to microSD cards, programming them with your favourite Raspberry Pi operating system has always been a little bit tricky," says Gordon.

> The utility reads the relevant file directly from our website and writes it straight to the SD card

Free to download

To get your copy of Raspberry Pi Imager, head to **magpi.cc/imager** and select and download the version of the imager for Windows, macOS, or Ubuntu. Raspberry Pi Imager will be auto-installed and launched on your computer.

Inside Raspberry Pi Imager, you'll find Raspbian as default operating system. Click on 'Raspbian (Other)' to choose Lite and Full versions of



Downloads

Raspbian is our official ope

Use **Raspberry Pi Imager** f

systems to an SD card read

- Raspberry Pi Imager f
- Raspberry Pi Imager f
- Raspberry Pi Imager f

Alternatively, use the links b

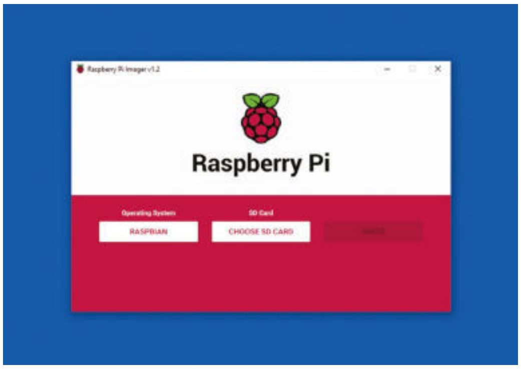
for our older installer for Ra

Raspberry Pi Imager for Windows, macOS, and Ubuntu make setting up your Raspberry Pi really simple

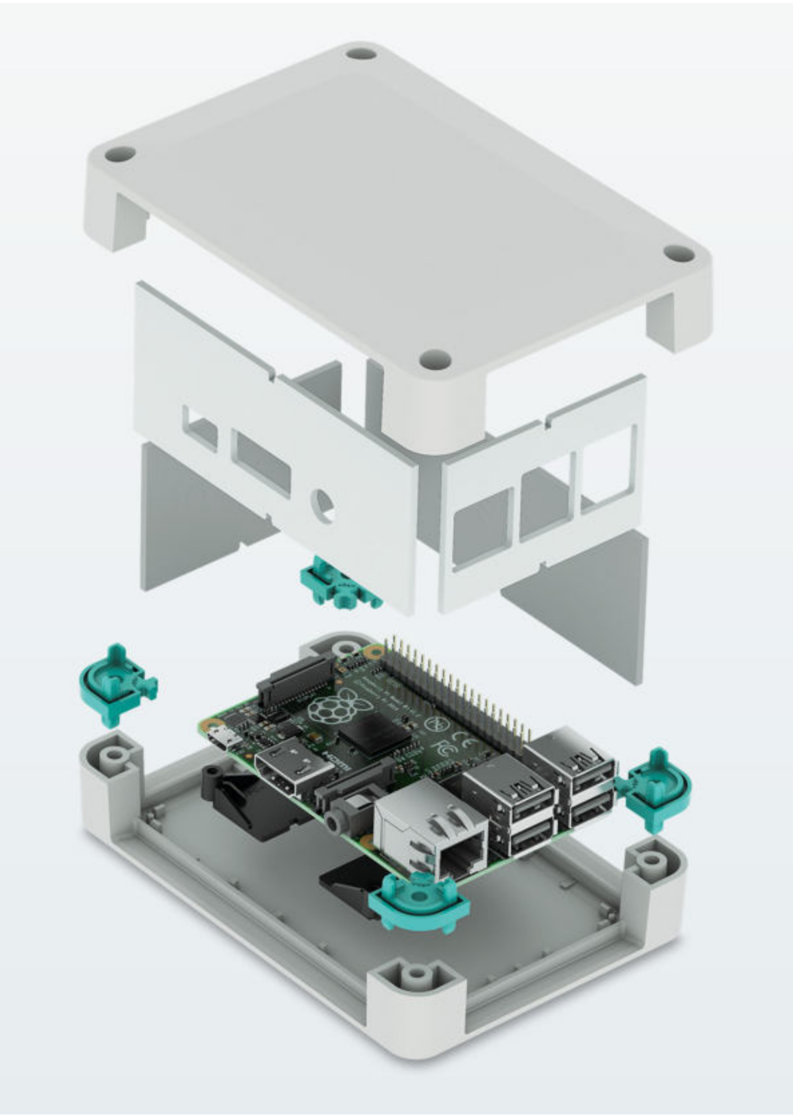
Raspbian. Other operating systems, such as LibreELEC and Ubuntu Server and Core, are also available. Raspberry Pi Imager includes a microSD card format tool and a Raspberry Pi 4 EEPROM boot recovery tool. You can also opt to install any operating system that you have downloaded as an image file.

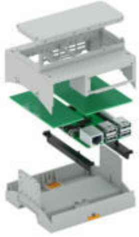
"Once you've selected an operating system from the available options, the utility reads the relevant file directly from our website and writes it straight to the SD card," explains Gordon. "During this process, Raspberry Pi Imager also caches the downloaded operating system image."

Clicking 'Write' initiates the OS writing process. Once complete, the microSD card can be inserted into the slot on Raspberry Pi and the OS launched as normal. M



Simply choose an operating system to install, then select your microSD card and write the OS to it



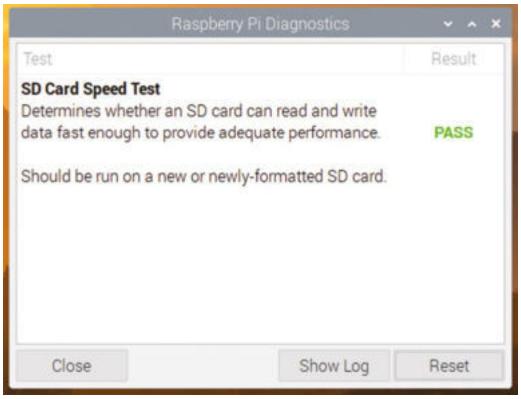


Raspberry Pi releases SD Card Speed Test

Raspberry Pi diagnostic tool lets you check your card's efficiency. By **Rosie Hattersley**

Raspberry Pi has launched its first diagnostic tool. The SD Card Speed Test tool measures the write and read speeds of the card in your Raspberry Pi. The utility gives a far more accurate indication of data throughput than the putative write/read capabilities claimed by card makers. A Raspberry Pi operating system runs from the microSD card on which it's stored, so the card's data transfer speed directly correlates to how fast programs and files can be loaded and read, as well as how fast data can be saved to the card.

> Measures the write and read speeds of the card in your Raspberry Pi



SD Card Speed Test
Determines whether an SD card can read and write data fast enough to provide adequate performance. PASS

Should be run on a new or newly-formatted SD card.

▲ Get an accurate measure of how fast your microSD card is transferring data to Raspberry Pi

Simon Long, Senior Principal Software Engineer at Raspberry Pi, says that while 'fake' SD cards are fairly easy to spot, "it's much harder to work out whether your supposedly fast SD card is actually meeting its specified speed."

The speed test tool is also useful for photographers and videographers because if the card can't keep up, image data won't be saved, leading to potential disappointment or even disaster (imagine if it were your honeymoon photos at stake or impossible-to-replicate dash-cam footage).

To install the new tool, open a Terminal window and enter:

```
sudo apt update
sudo apt install agnostics
```

Raspberry Pi will be launching more hardware diagnostic tools over the coming months. More information on the app can be found on the Raspberry Pi website: **magpi.cc/sdspeedtest**.

▶ microSD cards have a range of speeds, and it is important to have one that runs the Raspbian OS effectively

Instaclock

Designed to celebrate a new home, Instaclock uses two Raspberry Pi computers to great visual effect. **Rosie Hattersley** finds out more

MAKER

Riccardo Cereser

Riccardo is a UX designer for logistics company DFDS. After growing up in Italy and Dublin, he moved to Copenhagen to study interactive design. He's a keen illustrator and Raspberry Pi retro gamer.

@ricently

There's nothing like a deadline to focus the mind! Copenhagen-based illustrator and UX designer Riccardo Cereser was about to move into a new apartment with his girlfriend, and was determined his new home would have a unique timepiece. Instaclock (**magpi.cc/instaclock**) is the result.

Having studied at the Copenhagen Institute of Interactive Design, Italian-born Riccardo was keen that his new apartment would include an object that reflected his skills. He began sketching out ideas in Photoshop, starting with the idea of images representing numbers. "A hand showing fingers; a bicycle wheel resembling the number 0; candles on a cake; or the countdown numbers that appear in the beginning of a recording…" he suggests.

Having decided the idea could be used for an interactive clock, he quickly worked out how such an image-based concept might work displaying the hour, minutes, and seconds on displays in three wooden boxes.

Next, he set off around Copenhagen. "I started taking photos of anything that could resemble a number, aiming to create sets of ten pictures each based on a specific theme," he recalls. "I then thought how awesome it would be to be able to switch the theme and create new sets on the go, potentially by using Instagram."

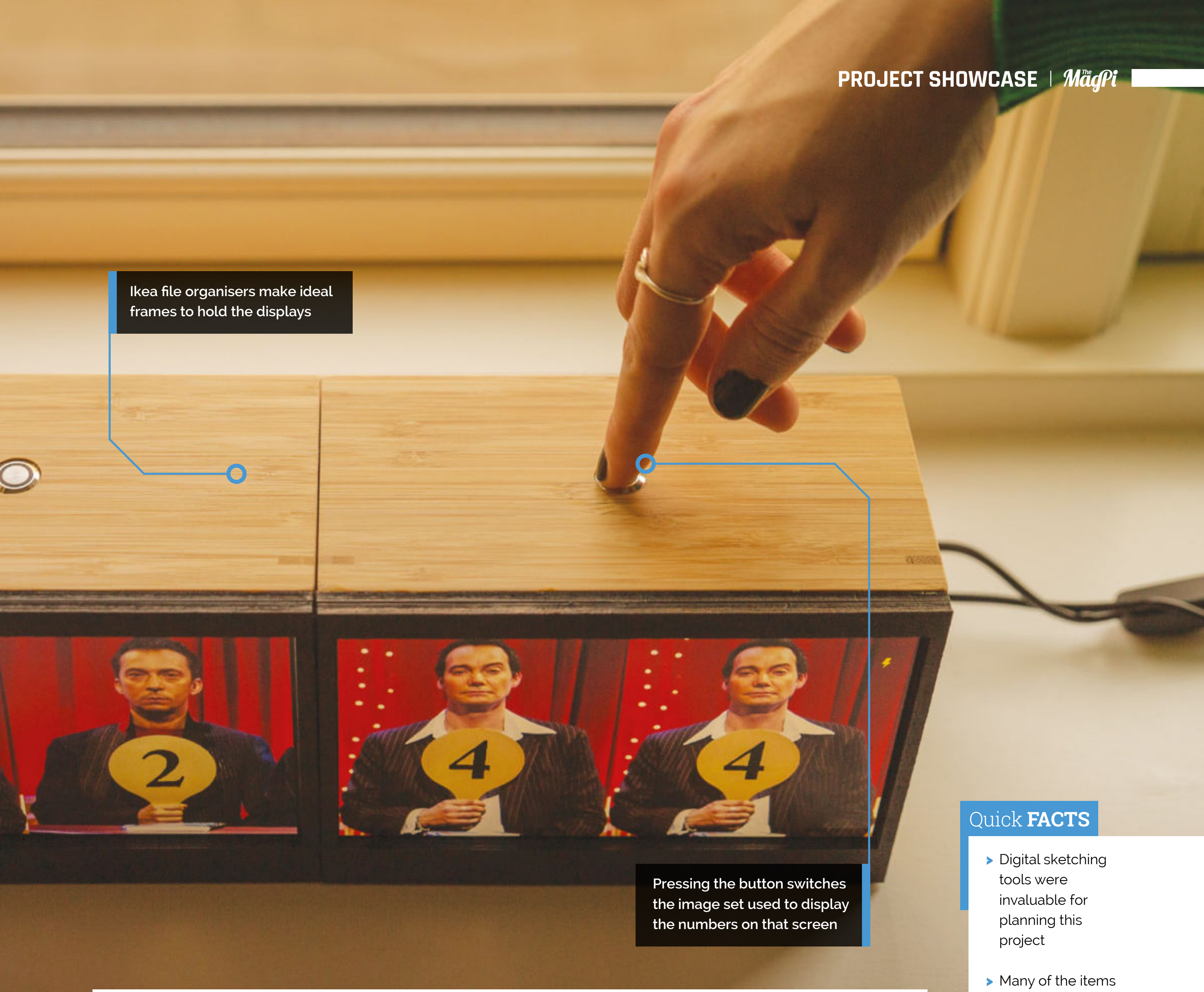
Instaclock displays fun images of numbers, rather than being a standard timepiece

This, Riccardo explains, is how the project became known as Instaclock. He was able to visualise his plan using Photoshop and made a prototype for his idea. It was clear that there was no need to display seconds, for example. Minute-by-minute updates would be fine.

> "I started taking photos of anything that could resemble a number"

Getting animated

Next up was figuring out how to call up and refresh the images displayed. Riccardo had some experience of using Raspberry Pi, and had even

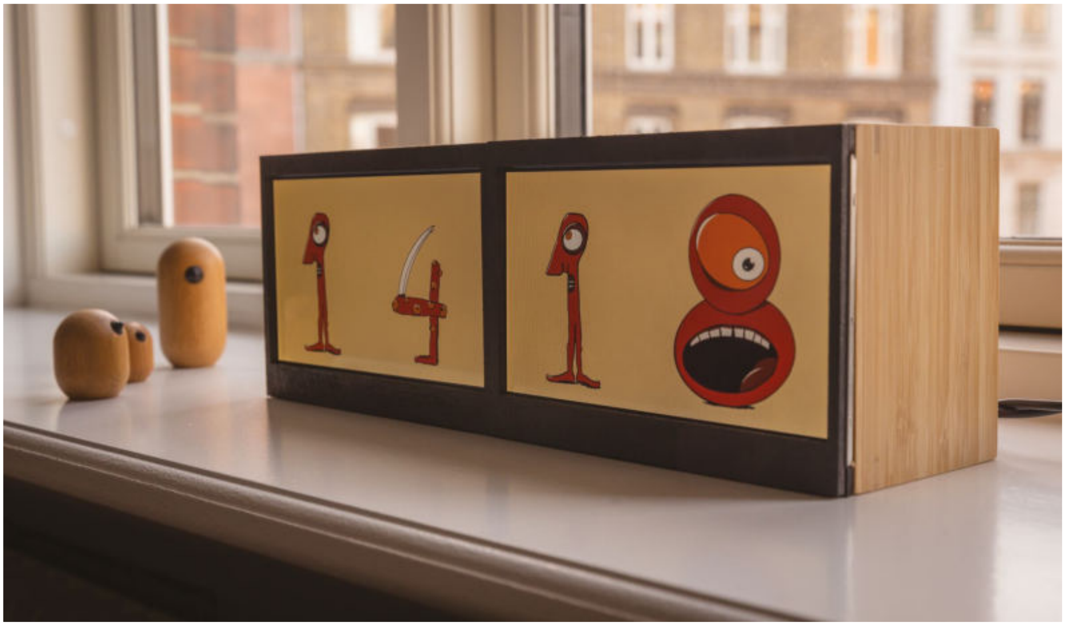


Ikea file organisers make ideal frames to hold the displays

Pressing the button switches the image set used to display the numbers on that screen

Quick FACTS

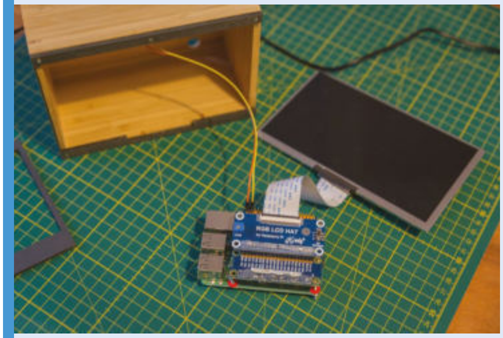
- Digital sketching tools were invaluable for planning this project
- Many of the items used to build Instaclock came from Ikea
- Riccardo got Instaclock's 'off switch' code from *The MagPi*
- He is thinking of adding an alarm function next
- He's pondering an animated weather cartoon version, too



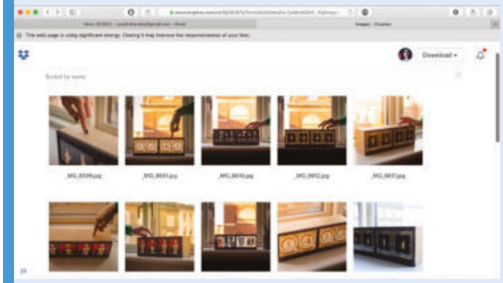
Stills from cartoons and illustrations work really well as a visual device and add to Instaclock's quirky appeal

Build a cool clock

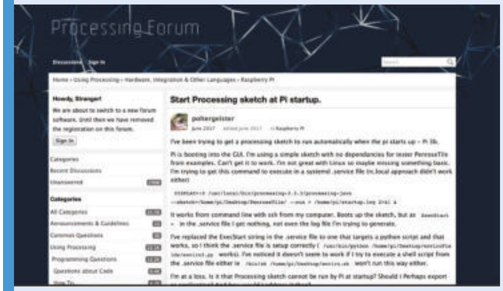
Each of the Instaclock's twin screens has a Raspberry Pi 3 Model B controlling image delivery to it. You'll need to adapt or build a box to house each screen, and drill holes to accommodate cables and ventilate it.



01 Attach the LCD HAT's ribbon cable to the screen's input port. You might also want to use magnets to help secure the screen in place in its display box. Make sure all the screens are grounded!



02 Install Raspbian on 8GB microSD cards, boot up, and connect each Raspberry Pi to your wireless network. Set a script to sync folders via Dropbox and use the cron scheduler to change the displayed image every two minutes (or another interlude).



Howdy, Stranger!

We are about to switch to a new forum software. Until then we have removed the registration on this forum.

Start Processing sketch at Pi startup.

poltergeister
June 2017 edited June 2017 in Raspberry Pi

I've been trying to get a processing sketch to run automatically when the pi starts up – Pi 3b.

Pi is booting into the GUI. I'm using a simple sketch with no dependancies for tester PenroseTile from examples. Can't get it to work. I'm not great with Linux so maybe missing something basic. I'm trying to get this command to execute in a systemd .service file (rc.local approach didn't work either)

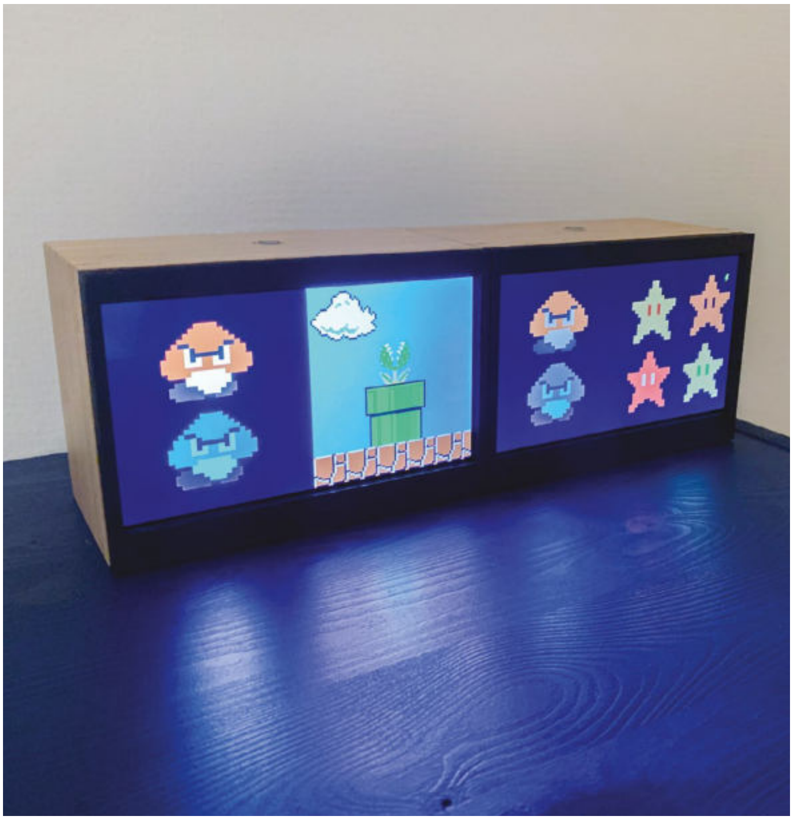
```
DISPLAY=:0 /usr/local/bin/processing-3.3.5/processing-java
--sketch=/home/pi/Desktop/PenroseTile/ --run > /home/pi/startup.log 2>&1 &
```

It works from command line with ssh from my computer. Boots up the sketch, but as ExecStart = in the .service file I get nothing, not even the log file I'm trying to generate.

I've replaced the ExecStart string in the .service file to one that targets a python script and that works, so I think the .service file is setup correctly (/usr/bin/python /home/pi/Desktop/[illegible] works). I've noticed it doesn't seem to work if I try to execute a shell script from the .service file either ie /bin/sh /home/pi/Desktop/[illegible].sh won't run this way either.

I'm at a loss. Is it that Processing sketch cannot be run by Pi at startup? Should I Perhaps export

03 Set the Processing sketch to launch at startup (see **magpi.cc/process**). Disable the screensaver (so it doesn't interrupt the clock script) and hide the cursor. Back up the microSD card, then switch on. Repeat steps for the other Raspberry Pi.



▲ Graphics from classic games may be used to tell the time – can you work it out?

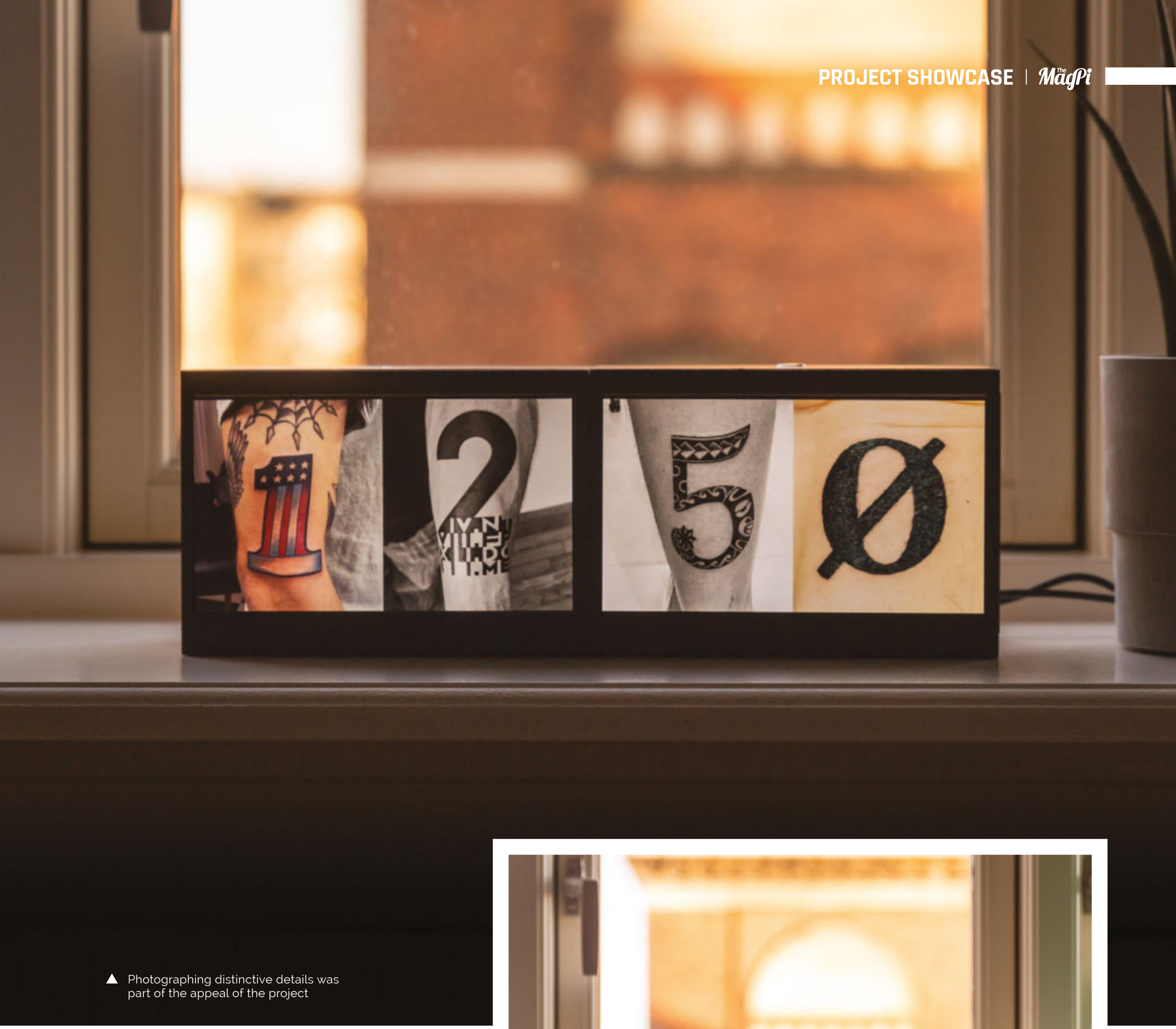
made a RetroPie games console. He also had a friend on the interactive design course who might just be able to help…

Creative coder Andreas Refsgaard (**andreasrefsgaard.dk**) soon got involved, and was quickly able to come up with a Processing sketch for Instaclock (**magpi.cc/instaclockgit**).

> " They also added a rule that shuts a screen down if the button on top of it is pressed for ten seconds or more "

Having spent dozens of hours looking into how an API might be used to pull in specific images for his clock, Riccardo was grateful that Andreas immediately grasped how it could be done. Riccardo then set parameters in cron for each Raspberry Pi used, so the Instaclock loaded images at startup and moved on to the next image set every ten seconds.

Because Riccardo wanted Instaclock to be as user-friendly as possible, they also added a rule that shuts



Photographing distinctive details was part of the appeal of the project

a screen down if the button on top of it is pressed for ten seconds or more. The script was one he got from *The MagPi* (**magpi.cc/offswitch**).

Assembly time

One of the most fun aspects of this project was the opportunity to photograph, draw, or source online images that represent numerals. It was also the most time-consuming, of course. Images reside in Dropbox folders, so can be accessed from anywhere. Deciding on a suitable set of screens to display them, and boxes or frames for them, could also have dragged on but for an impromptu visit to Ikea. Riccardo fortuitously found that the Waveshare screens he'd selected would fit neatly into the store's Dragan file organiser boxes. He was then able to laser-cut protective overlays secured with tiny magnets.

Visitors to Riccardo's apartment have been intrigued and challenged by the displays, which even feature the occasional maths problem

CNC Etch-A-Sketch

Engineer Quint Crispin has been drawing lots of attention to his latest project, as **David Crookes** discovers

MAKER

Quint Crispin

Quint is an engineer who loves to play around with cool builds. He created his Etch-A-Sketch project with the help of his young son, who assisted with its fine-tuning.

magpi.cc/cncetch

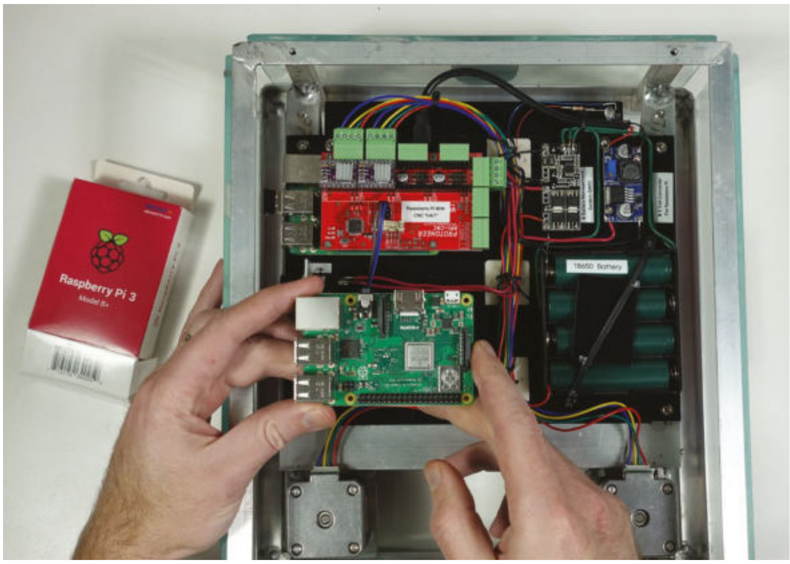
The project is powered by four lithium cells in conjunction with a battery management system to give five hours of operation

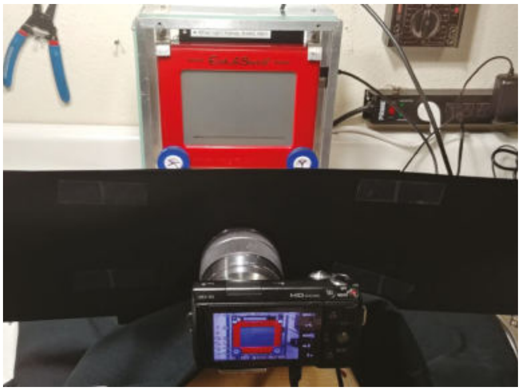
For 60 years, the iconic Etch-A-Sketch toy, invented by French electrician André Cassagnes, has provided a blank canvas that has both enthralled and infuriated. Many of us will go no further than drawing some simple straight lines and shapes. But there are some amazing practitioners out there (check out Jane Labowitch and Dave Roberts) who can generate incredible works of art.

Quint Crispin admits he is not one of them, but, with a bit of ingenuity, the engineer has nevertheless found a way to unleash his creativity. He has devised a machine to perform the magic for him, using a Raspberry Pi 3 computer connected to a CNC HAT made by Protoneer. This has enabled two NEMA-17 size stepper motors to be controlled – hooked, as they are, to the Etch-A-Sketch's two knobs using a belt attached around blue timing pulleys.

Shaking things up

"I had no idea the Raspberry Pi was so powerful," he says. "But when I discovered it was capable of driving stepper motors using programs such as bCNC, I knew I'd found my solution for a portable CNC Etch-A-Sketch." Indeed, his use of our favourite computer meant he could produce a wireless unit that was so light and sturdy, it could also be picked up and shaken to clear the display without causing any damage.





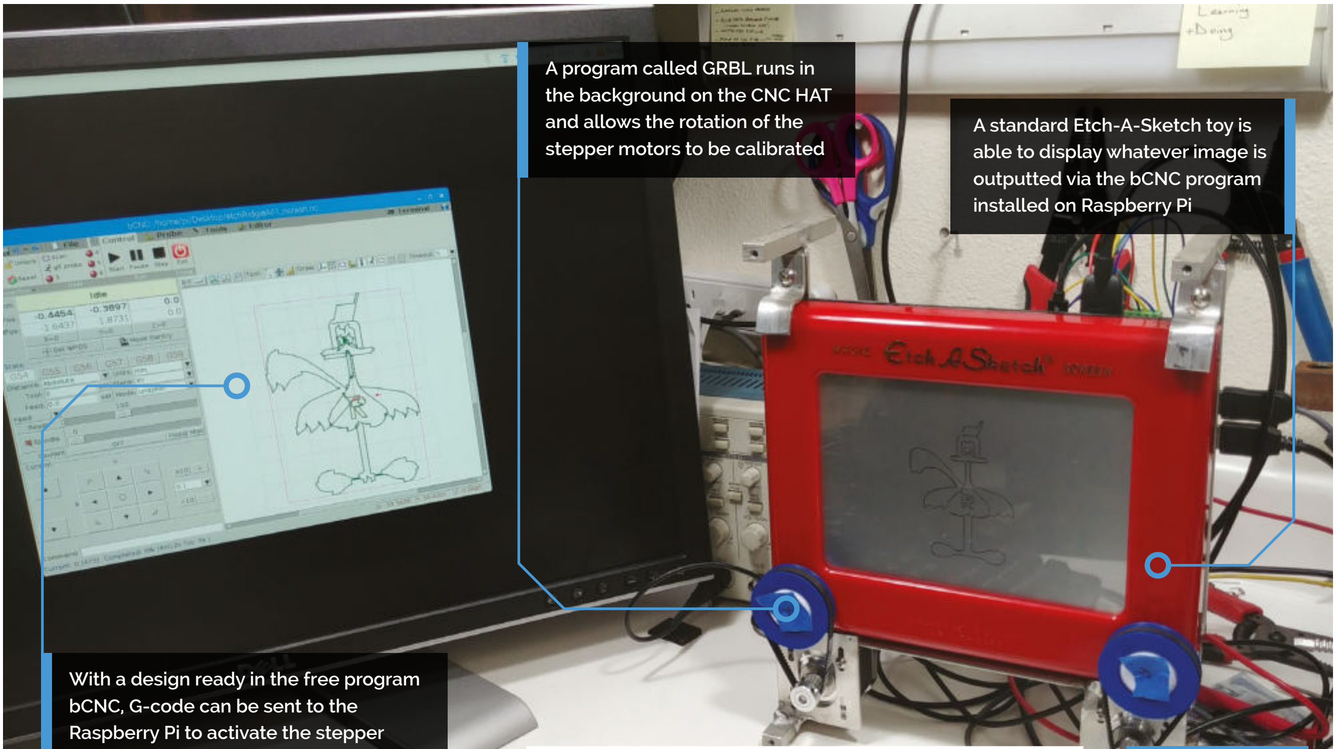
For the creation of stop-motion videos, the camera sits behind black fabric to prevent reflections, and each image is drawn 1 to 2 mm apart for smooth footage

"Other projects have connected stepper motors to an Etch-A-Sketch, but the makers have quickly found they have to take the whole thing apart just to reset the screen," Quint tells us. "That's something I've never had to worry about because I spent a lot of time designing and producing a strong frame. It meant more work up front, but was well worth it in the long run."

Even so, the challenge didn't stop there. As with all CNC machines (those which use software connected to a tool to automate control such as 3D printers), simple x and y co-ordinate commands are issued. "You're basically telling the machine 'go here, go there, now here, now there' using commands known as G-code," Quint explains. Coding is, therefore, crucial.

Turning things around

For this, Quint used variations of the G-code command G1, noting the required x/y position and the required speed. "As an example, G1 X40 Y60



A program called GRBL runs in the background on the CNC HAT and allows the rotation of the stepper motors to be calibrated

A standard Etch-A-Sketch toy is able to display whatever image is outputted via the bCNC program installed on Raspberry Pi

With a design ready in the free program bCNC, G-code can be sent to the Raspberry Pi to activate the stepper motors to create it on the Etch-A-Sketch

Quick FACTS

- The project cost less than $200
- It uses 3D printer stepper motors
- The frame is light enough to shake
- Raspberry Pi is running bCNC
- It also wears a CNC HAT

> "Using small enough line segments, you can even draw a circle!"

F400 means go 40 mm to the right and 60 mm up at a speed of 400 mm per minute," he says. "Using small enough line segments, you can even draw a circle!"

The first real problem he encountered was the looseness in the Etch-A-Sketch mechanism when changing directions. "You have to turn the wheels further than you think – if you mark an Etch-A-Sketch dial and bring it to the same spot from both directions, you'll get about a 1.5 mm mismatch which can make your sketches look really poor. Most CNC software has 'backlash compensation' to account for this, but bCNC does not. Fortunately, I stumbled on a free program called NoLash that adds backlash compensation to the G-code program itself. You just tell it how much to compensate for and it does the job for you."

Now, by firing up bCNC, Quint can create an image and have it replicated on the Etch-A-Sketch screen with the knobs being automatically manipulated. It has also allowed for some very elaborate designs, which have impressed all who have seen them being created. "Once I got it figured out, everything worked really well," he says. "The experience has been magical."

For fun, Quint decided to play around with stop-motion video. He'd create an image, take a photo of the Etch-A-Sketch screen, shake, and create another image

Solar-powered Raspberry Pi Camera

Kaspars Dambis has put an eye in the sky using his Raspberry Pi, as **David Crookes** explains

MAKER

Kaspars Dambis

Kaspars is a web developer and electronics enthusiast who enjoys creating simple, low-power connecting projects that use microcontrollers.

kaspars.net

Building a new family home from scratch is not for the faint-hearted. Neither, on reflection, is climbing up a tree to install a solar panel and a CCTV camera, and yet Kaspars Dambis has done both.

One thing led to another. "I wanted to monitor the construction site remotely because we currently live 100 kilometres away," he says. But rather than buy an off-the-shelf CCTV system, he purchased a dummy camera case, creating his own way of capturing images using a Raspberry Pi 3 Model B computer.

"From the start, I wanted to keep watch over the building of our home and create a time-lapse of the process for fun," explains Kaspars. "I knew it had to be a battery-powered and wireless system because getting mains power to the device wasn't going to be possible on a site like that."

Seeing the light

Kaspars picked up a lightweight 18 V 5 A solar panel that was marketed as being perfect for charging boats and cars. This, he figured, would gather energy from the sun to charge a 12 V battery and, with the use of an inexpensive 12 V-to-5 V buck module, power the Raspberry Pi 3 Model B and an eight-megapixel Raspberry Pi Camera Module v2.

Fitting the system in a tree prevents tampering and makes the setup look less intrusive

At first, Kaspars attempted to build a case from a generic project box, but he found it was relatively difficult to find decent clear glass for the camera window. "I then noticed a dummy security camera in a local hardware store which had perfect clear glass on one end, and full weather projection for the battery-powered LED system," he says. And this worked a treat.

> "The weather protection proved to be great, even during the winter months"

Using hook-and-loop fasteners, the Camera Module could be attached to Raspberry Pi's case which, in turn, was secured to the inner housing of the camera casing. "The weather protection proved to be great, even during the winter months, and [my] Raspberry Pi never had issues with the temperature ranges either."

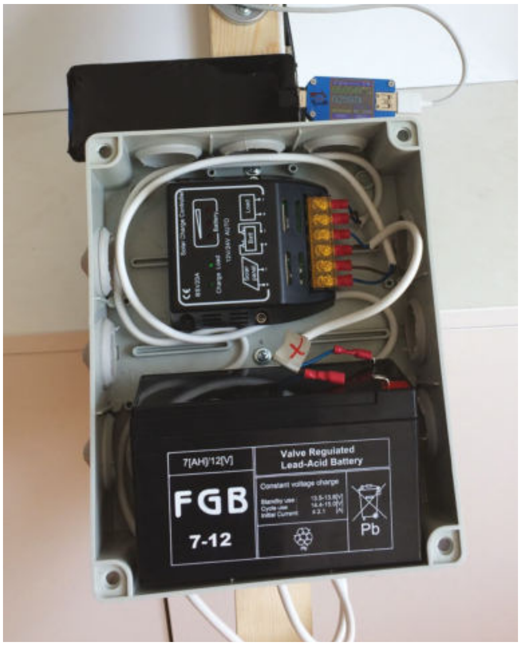
To get as much energy as possible, the solar panel and the 12 V battery were positioned up high on a wooden frame, with the security camera device attached too. This wasn't ideal. "The battery was heavy and the frame broke during a storm, so I created two separate frames: one for the solar panel with the camera unit below and another for the battery and the solar charge controller," he says.

Monitoring from afar

Kaspars configured his Raspberry Pi to connect to a nearby WiFi access point via a standalone

Alert! Batteries included

Please take care with rechargeable car batteries: **magpi.cc/carbat**

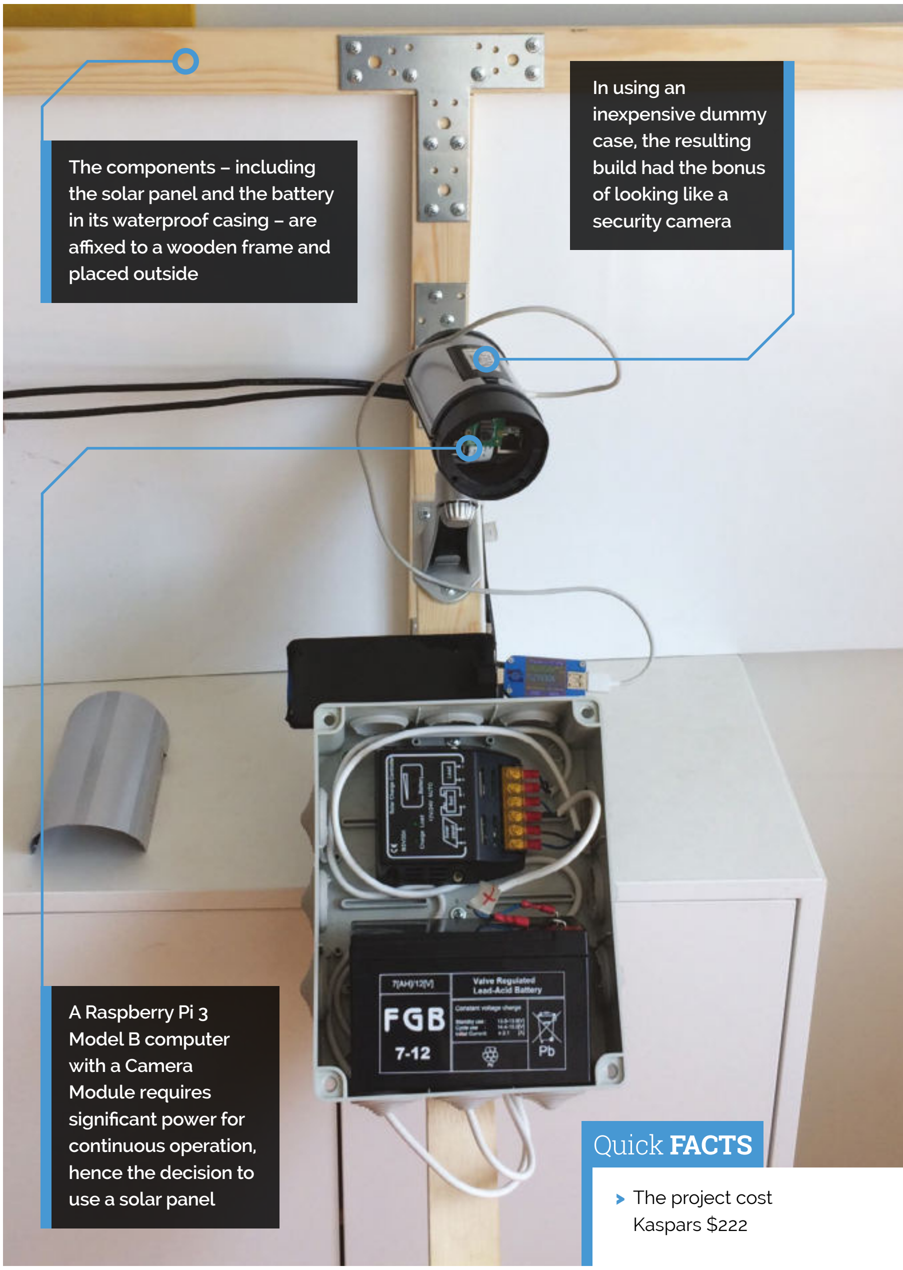


A 12 V, 7 Ah lead-acid battery has been sufficient to provide an average power consumption of 2 W since it can provide 84 watt-hours of energy

modem plugged in at his neighbour's house by setting the access credentials in the **wpa_supplicant.conf** file in the root of the SD card. "The most basic time-lapse functionality was added using a simple Bash script which takes a picture from the raspistill tool at the desired time intervals and stores it on the SD card."

Connecting remotely involved setting up an external server and using the SSH tunnel, autossh. When up and running, a photo is taken every hour and it can be downloaded using SCP when connected to the same WiFi access point as Raspberry Pi. Video capture and real-time feeds can also be viewed in a browser.

It all means Kaspars is now able to remotely log in to his Raspberry Pi and eventually see his new home come into being. There is still an element of fear, though. "The ability to connect to a Raspberry Pi which is up in a tree powered by the sun and 100 km away is a special feeling," he says. "Every command you type in the terminal has the potential to break the WiFi connection – and the cost of each mistake is a 200 km drive to restart the device."



The components – including the solar panel and the battery in its waterproof casing – are affixed to a wooden frame and placed outside

In using an inexpensive dummy case, the resulting build had the bonus of looking like a security camera

A Raspberry Pi 3 Model B computer with a Camera Module requires significant power for continuous operation, hence the decision to use a solar panel

Quick FACTS

- The project cost Kaspars $222
- He's also paying for 10GB of monthly data
- It should run for two days without charging
- Getting it up a tree was the hardest part
- All cases need to be waterproof

Dummy security cameras can be picked up cheaply – this one cost Kaspars just $8

Flying Saucer Clock and NAS

If you're thinking of fashioning yourself a new clock, have you considered one with an extraterrestrial twist? **Nicola King** has a close encounter...

MAKER

Harry Tansey

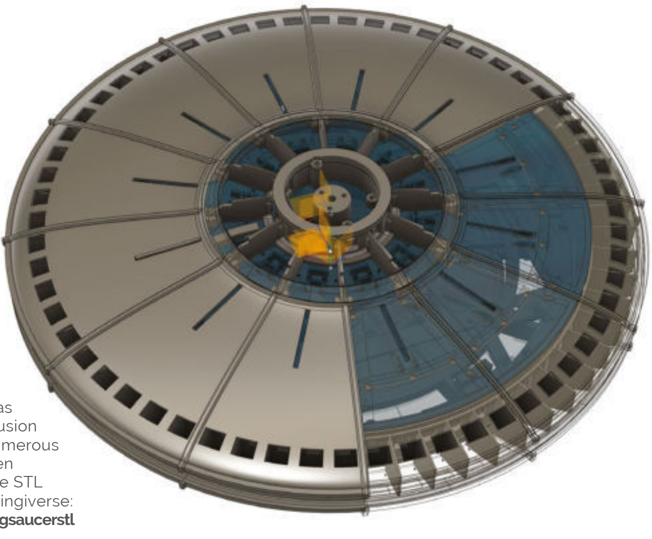
A software engineer, DIYer, and maker, with interests in music visualisation, XR, 3D design and printing, Raspberry Pi, microcontrollers, electronics, and blinky LEDs.

planetzarg.com

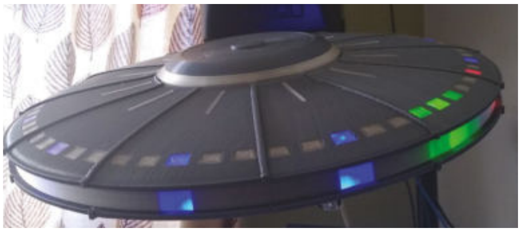
UK-based Harry Tansey is a maker who'd been considering creating some form of clock project for a while, when he was inspired to make something that did more than just tell the time. "I have been experimenting with WS2812B LEDs (aka NeoPixels) over the past few years and knew they were going to be a key component," he says. "When my NAS (network-attached storage) box started playing up last year, I knew I had to make or buy a replacement. Given that both a NAS and a clock tend to be on 24/7, I started toying with the idea of ring of 60 LEDs as clock segments and arranging the NAS components (Raspberry Pi, HDDs, and PSU) inside."

Print perfect

So, the concept for the Flying Saucer Clock was born, and Harry set about designing a prototype using Fusion 360. Given that he was going to 3D-print the clock, he knew it was critical to get the design as precise as possible, in order to save both time and money with expensive, failed 3D prints. "Failure can be expensive when the weight of filament in the project is just short of 2 kg! This equates to nearly 120 hours of printing time – or about four days using two 3D printers running simultaneously for around 14 hours a day," he calculates.



▶ The saucer was designed in Fusion 360 and its numerous parts were then 3D-printed; the STL files are on Thingiverse: **magpi.cc/flyingsaucerstl**

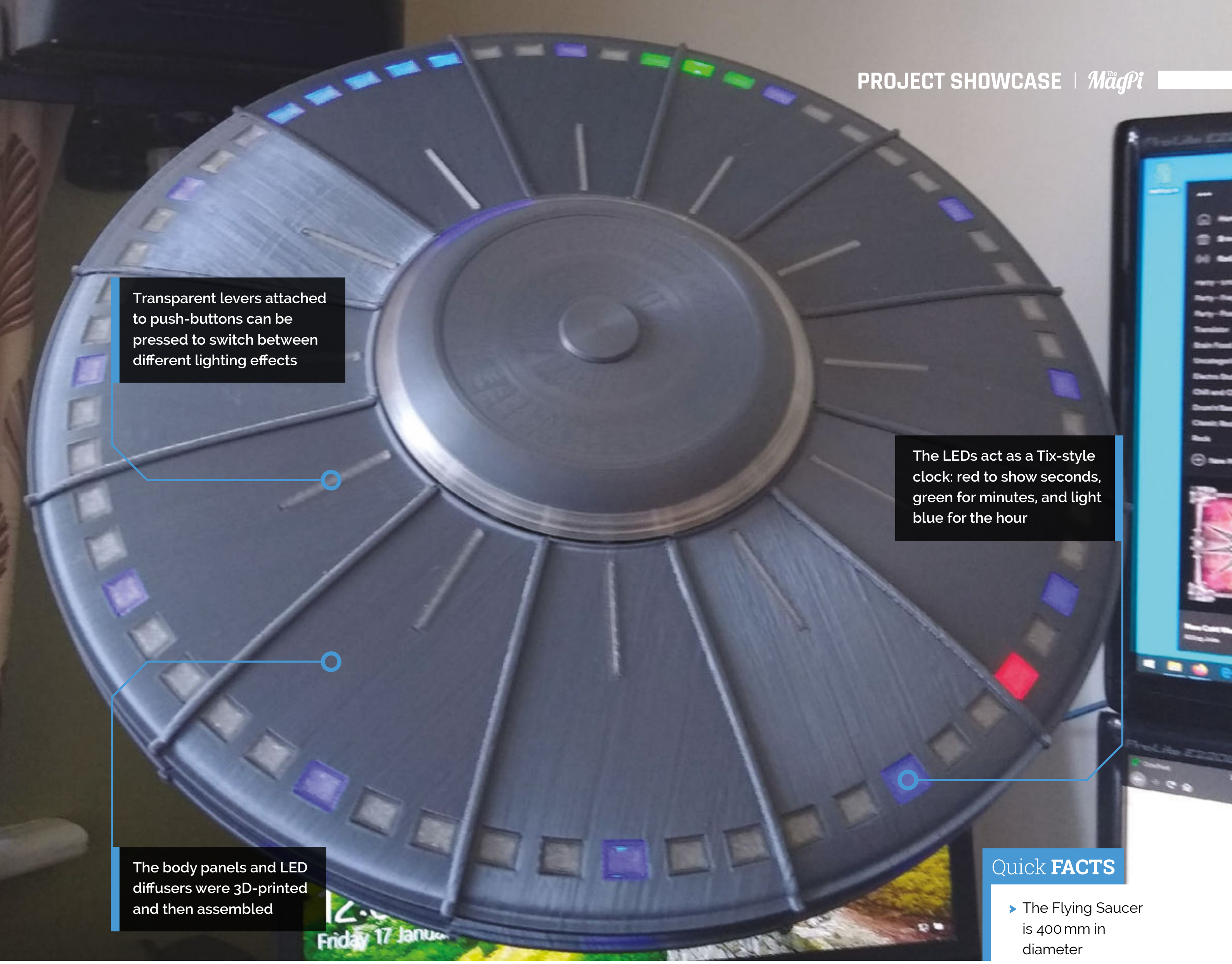


▲ A side view of the saucer. Transparent segments are used to diffuse the LEDs, with light blockers between them for crisp separation

Harry also set himself another challenge: "It was critical to give the pieces enough tolerance so they could be used straight from the printer and fit together with minimal surface finishing."

Despite a few hiccups with his first attempt, Harry pushed on, and the finished article is something to behold. So, how is it constructed? "The outer ring has 60 RGB LED segments constructed simply by cutting a length of a standard WS2812B LED strip and mounted in a 3D-printed holder with protrusions to keep the LEDs fixed in place," he says. "Lighting one of outer LEDs red indicates the seconds, and a green one indicates the minutes. I found that lighting the two LEDs to either side of the minute LED improved the readability."

Harry also created an inner ring of LEDs divided into twelve segments, and one of these is lit in blue to indicate the hour. He then lit up some hour markers on twelve of the outer 60 LEDs, making it easier to track the positions of the minute and second LEDs. Harry chose a Raspberry



Transparent levers attached to push-buttons can be pressed to switch between different lighting effects

The LEDs act as a Tix-style clock: red to show seconds, green for minutes, and light blue for the hour

The body panels and LED diffusers were 3D-printed and then assembled

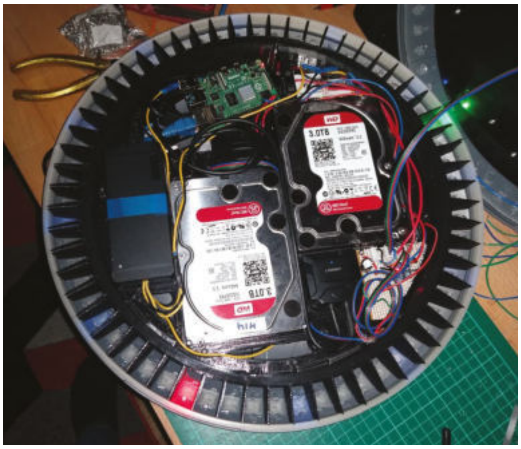
Pi 4 to drive the LEDs as he was "really excited by the capabilities of [its] full-speed Ethernet and USB 3 ports."

The Flying Saucer's NAS file-sharing functionality comes by way of two 3.5-inch drives, secreted inside, and a fairly simple Samba setup. "I have created different file share areas for work, family members, and storage for my Linux-based satellite TV/PVR receiver," reveals Harry.

Udderly ingenious!

Harry says that the feedback from family and friends has been very positive, but, he's not finished there with his flying saucer. "There is another ring of twelve RGB LEDs on the bottom of the UFO. These all face inwards and the final effect runs these in a slow pulsating pattern. With a semi-transparent cup that I borrowed from my daughter, it makes a really convincing beam effect that appears to 'travel' down the cup as the light intensifies. I am going to use this together with a 3D model of a cow for the 'traditional' cow abduction effect."

> "Failure can be expensive when the weight of filament in the project is just short of 2 kg!"



The two NAS disk drives and Raspberry Pi are housed within the saucer, powered by a 12 V PSU via three buck converters

Quick FACTS

- The Flying Saucer is 400 mm in diameter
- It's lit up by a total of 84 NeoPixels
- The twelve body panels were printed in vertical orientation
- See how Harry made it here: **magpi.cc/flyingsaucernas**
- Harry has around twelve Raspberry Pi boards around the house, including one controlling his central heating

Chamber: Sourdough Incubator

Having issues with your sourdough starter? Read on and be inspired… here's a radical baker using his loaf. **Nicola King** delves into the dough

MAKER

Trent Fehl

US-based Trent is a maker with electrical engineering training and a love for writing software. His personal projects are often inspired by his experiences outside of tech – most often cooking, hiking, and cycling.

magpi.cc/chamber

When maker, and baker, Trent Fehl first started making sourdough bread, he realised that keeping the starter at the correct, steady temperature could be a bit of a minefield. "I read all about different techniques for keeping a culture alive – of the different conditions to monitor, temperature was chief," he tells us. "Bakers have all sorts of tricks to achieve that temperature. Some have a spot on their kitchen counter-top they like, others prefer an oven (turned off) with the light on, but many recommend specialised proofing boxes. What struck me was that only one of these options allowed you to set the temperature and you could only control to temperatures higher than ambient."

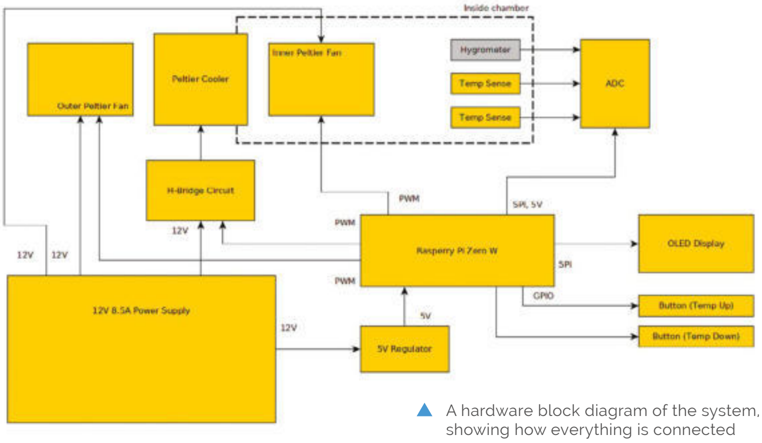
All you knead to know

Trent decided to take on the challenge of designing a product to meet his culinary needs, and was sure that he wanted to incorporate a small Peltier cooler in his project, as his friend Scott Hutchinson had given him the idea on a camping trip. "He happened to be a spacecraft thermal systems engineer and suggested using the Peltier cooler for both cooling and heating," says Trent. "I thought this was so slick that I really got moving on the project when I got home."

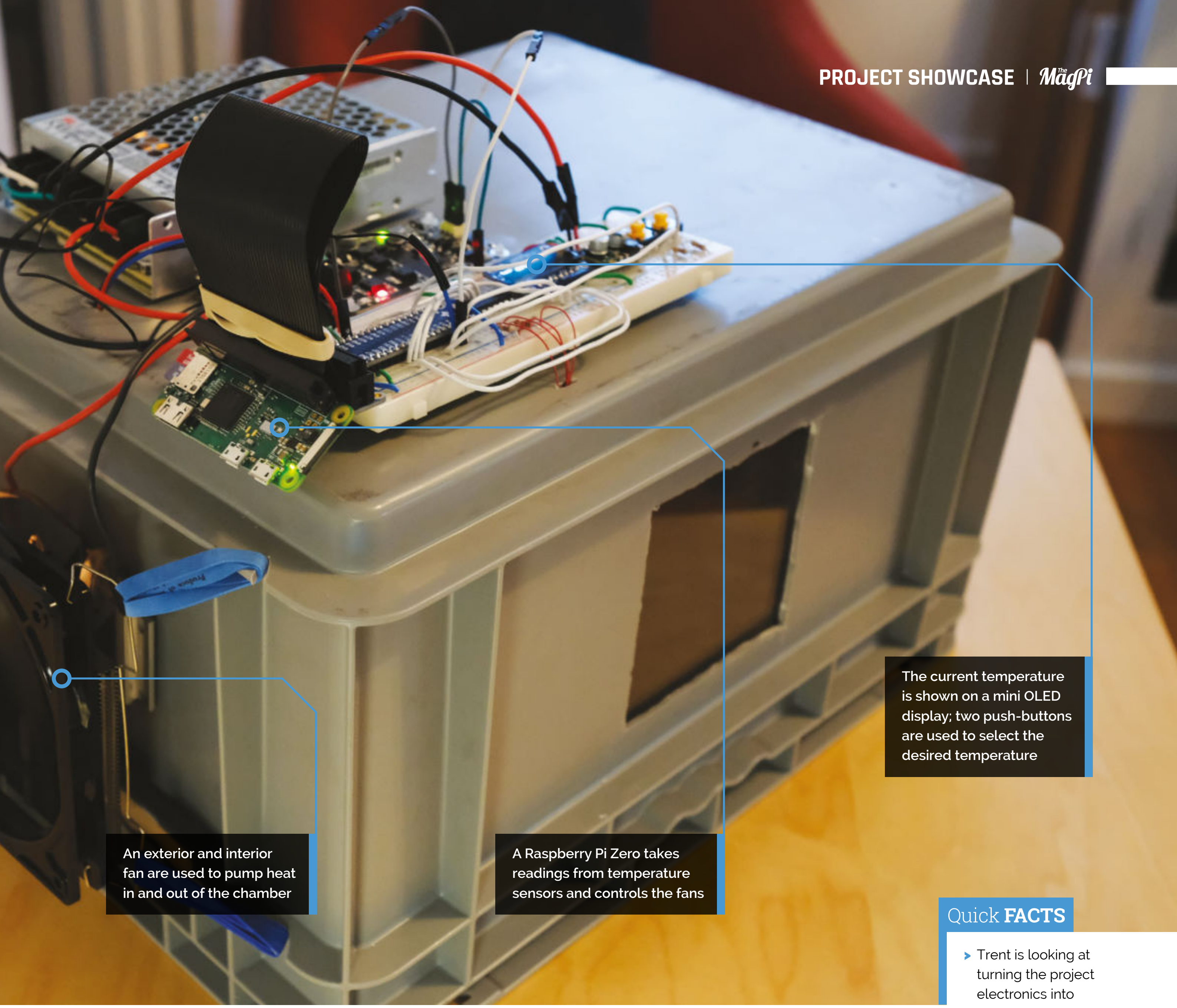
The fruit of Trent's labours: a delicious sourdough loaf

So, how exactly does the incubator work? "The Chamber utilises a Peltier cooler, also known as a thermoelectric cooler, to either pump heat from inside the Chamber to the outside (cooling the interior) or to pump heat from outside the Chamber to the inside (heating the interior)," explains Trent. "The direction that the heat is pumped is simply controlled by alternating the polarity of the voltage applied to the Peltier cooler."

He changes the temperature in the Chamber with an H-bridge module driven by a Raspberry Pi Zero. So, if the temperature gets too high, the fan on the outside wall pushes the heat away, and if too cold, another inside fan pulls warm air in. This being Trent's first Raspberry Pi project, he appears converted, saying the single-board computer "is



A hardware block diagram of the system, showing how everything is connected



The current temperature is shown on a mini OLED display; two push-buttons are used to select the desired temperature

An exterior and interior fan are used to pump heat in and out of the chamber

A Raspberry Pi Zero takes readings from temperature sensors and controls the fans

just such a great tool for personal projects: there is an excellent community offering software libraries, lots of compatible hardware, and helpful guides."

You say sour, I say sauer

It's true that bakers can buy proofing boxes, but Trent thinks that his incubator has clear advantages over commercially available alternatives. "The big advantage my chamber offers is heating and cooling in one package; plus it might be cheaper. Some of the off-the-shelf, heat-only options are almost $200 new, while mine is ~$180." In addition, Trent's invention can be used for other foodstuffs: "I enjoy baking, and I really enjoy eating bread, but I'd be lying if I didn't say that the mad scientist aspect of harnessing the power of fungi and bacteria to create tasty foods didn't draw me in... we've also fermented jars of garlic and jalapeños, sauerkraut, and various peppers. I'm proud to report that everything has tasted good."

> " I realised there were more people with this problem than I originally thought "

Trent demonstrated his Chamber at the 2019 Hackaday Superconference, and got a brilliant response from like-minded makers. "Right at the start of my talk I said something like, 'for those of you who maintain sourdough starters, you might be familiar with specific target temperatures but no means to control to those temperatures,' and I noticed that a good number of people were smiling and nodding their heads. At that moment, I realised there were more people with this problem than I originally thought," he says.

Quick FACTS

- Trent is looking at turning the project electronics into a HAT
- The Chamber can keep a steady temperature between 16 and 28°C
- The software for the project is written in C++
- See the GitHub page for code and details: **magpi.cc/chambergit**
- Trent is experimenting with machine learning projects using Raspberry Pi

PiVidBox

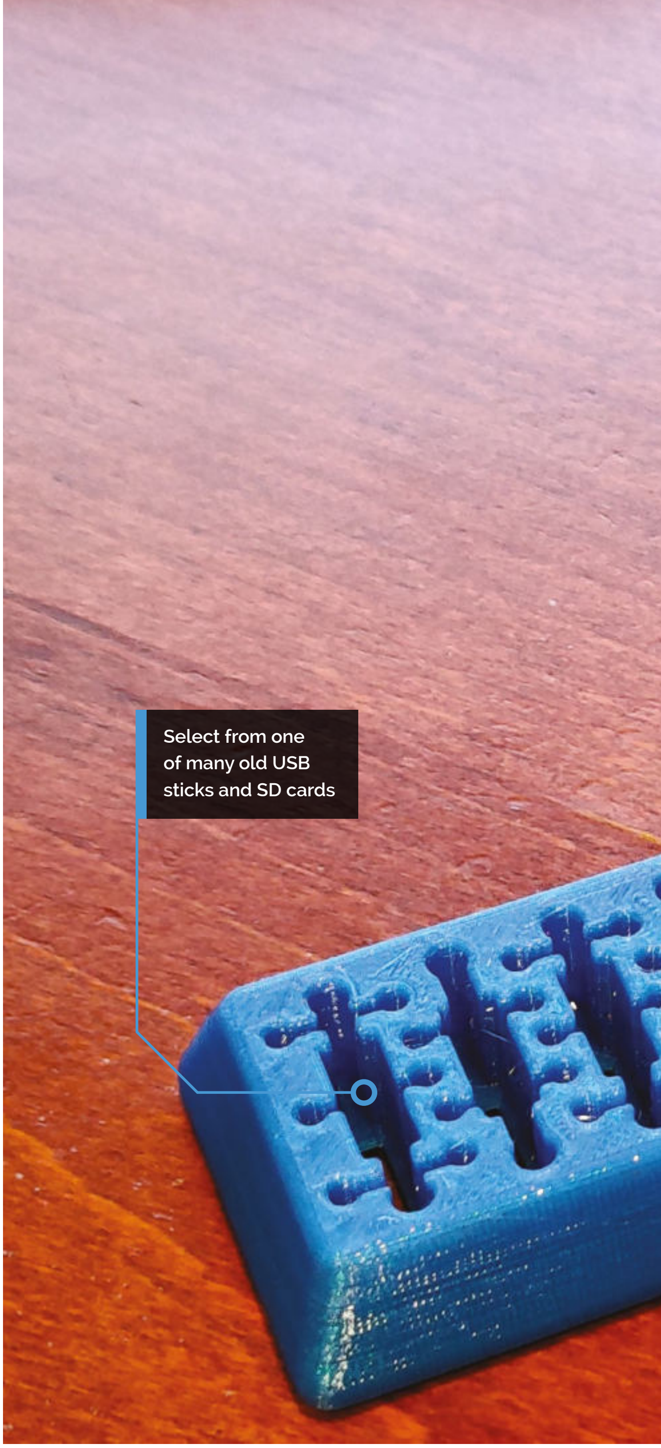
Recycling tech in a different way so that young kids can use it. **Rob Zwetsloot** checks it out

MAKER

Roiy Zysman

A software developer and product manager now running startup ledgerlets.com. He's used Raspberry Pi in many classic projects.

magpi.cc/pividbox



Select from one of many old USB sticks and SD cards

When it comes to recycling tech, we usually think of upcycling and the excellent work on Martin Mander. Roiy Zysman took a different approach, though, with his PiVidBox.

"My project is a simple-to-use Raspberry Pi based media centre that even children as young as three years old can use," he tells us. "It provides a simple, physical interface that's based on old and most likely discarded hardware. It's simple, because instead of a full-blown graphical interface that may be too complicated for small kids to operate, it uses a physical interface that relies on a simple action of plugging in USB thumb drives to Raspberry Pi."

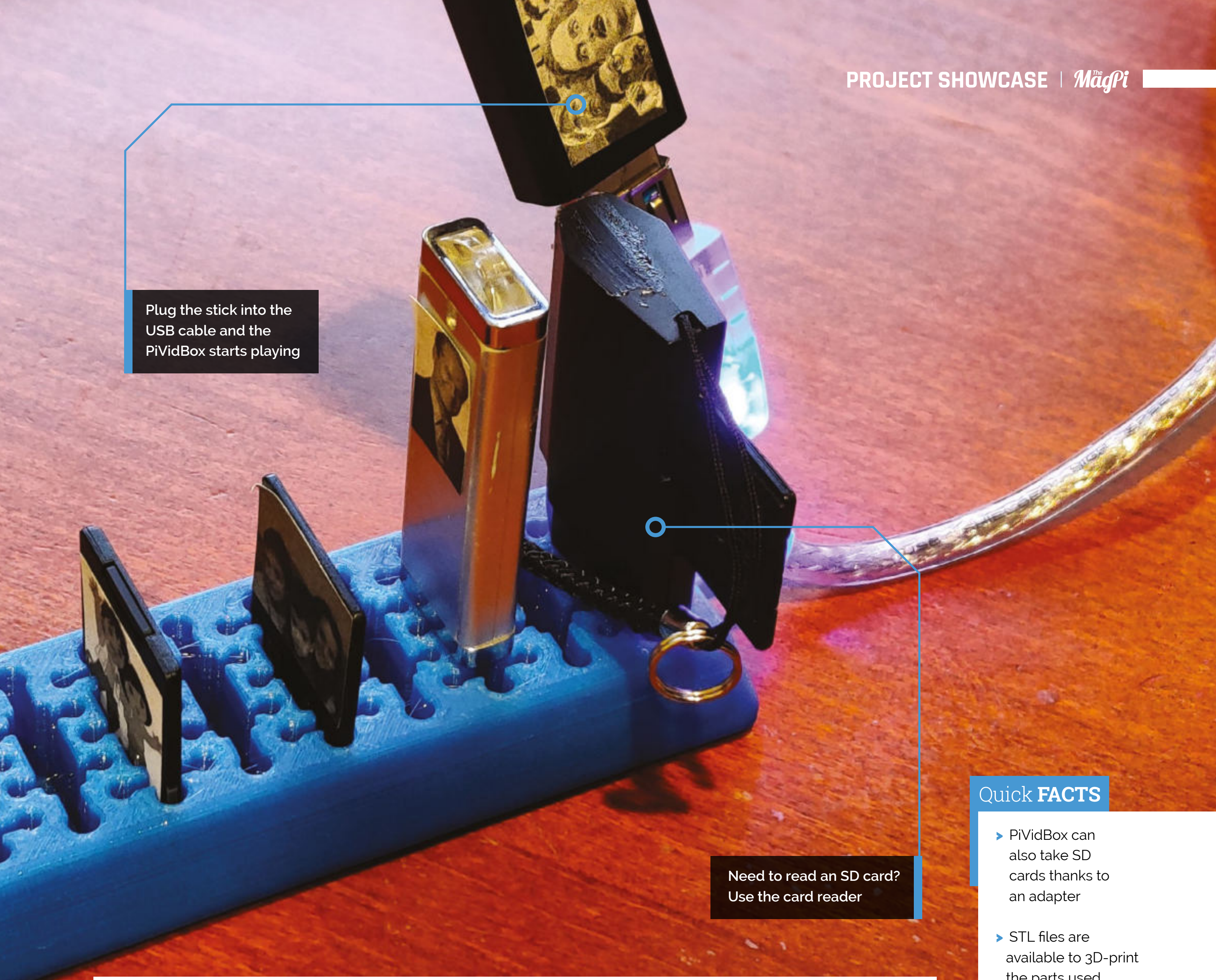
> We have thumb drives with cartoons, anime, and youth shows for our kids

It's a quite ingenious bit of low-tech design, and anyone who has seen a toddler work a DVD player recently will understand how physical interaction is something they can understand.

The system works by having the USB drive checked by Raspberry Pi during boot, and then playing a random video from it. You can easily swap it out for other USB drives, as the script can tell: "We have thumb drives with cartoons, anime and youth shows for our kids, but we also made a thumb drive with our old favourite shows when me and my wife want to watch some reruns of nostalgic shows," says Roiy.

Empowering kids

"Having three kids at home, I wanted to empower them by giving them the ability to choose what video content they want to watch by themselves," he explains. "But on the other hand, I also wanted to control what kind of content they consume rather than letting them roam freely on video streaming services such as Netflix and YouTube. I also had a

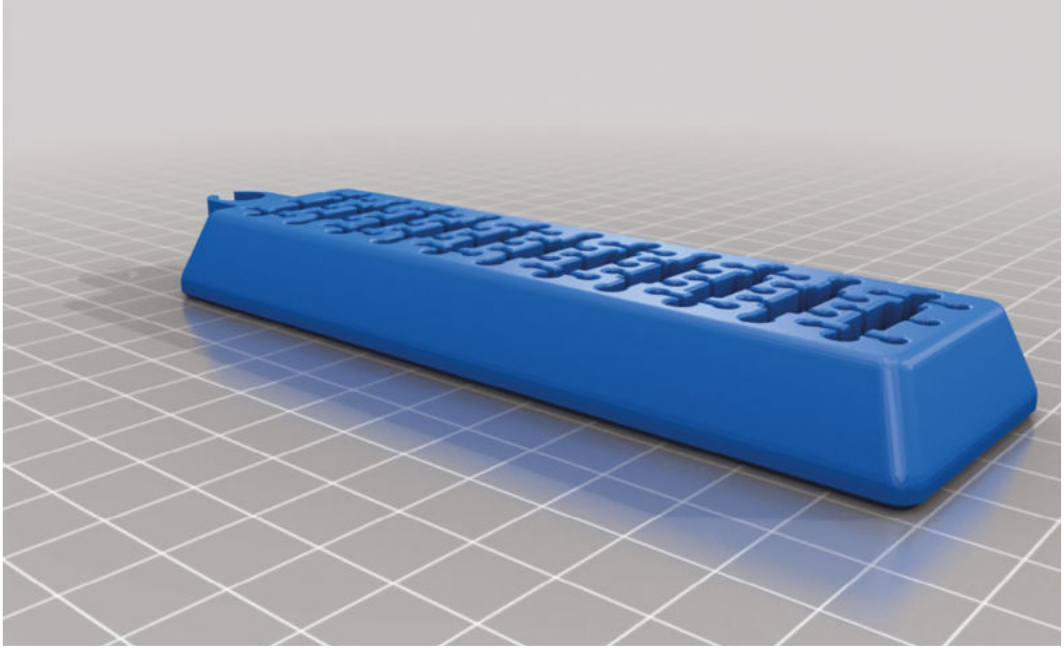


Plug the stick into the USB cable and the PiVidBox starts playing

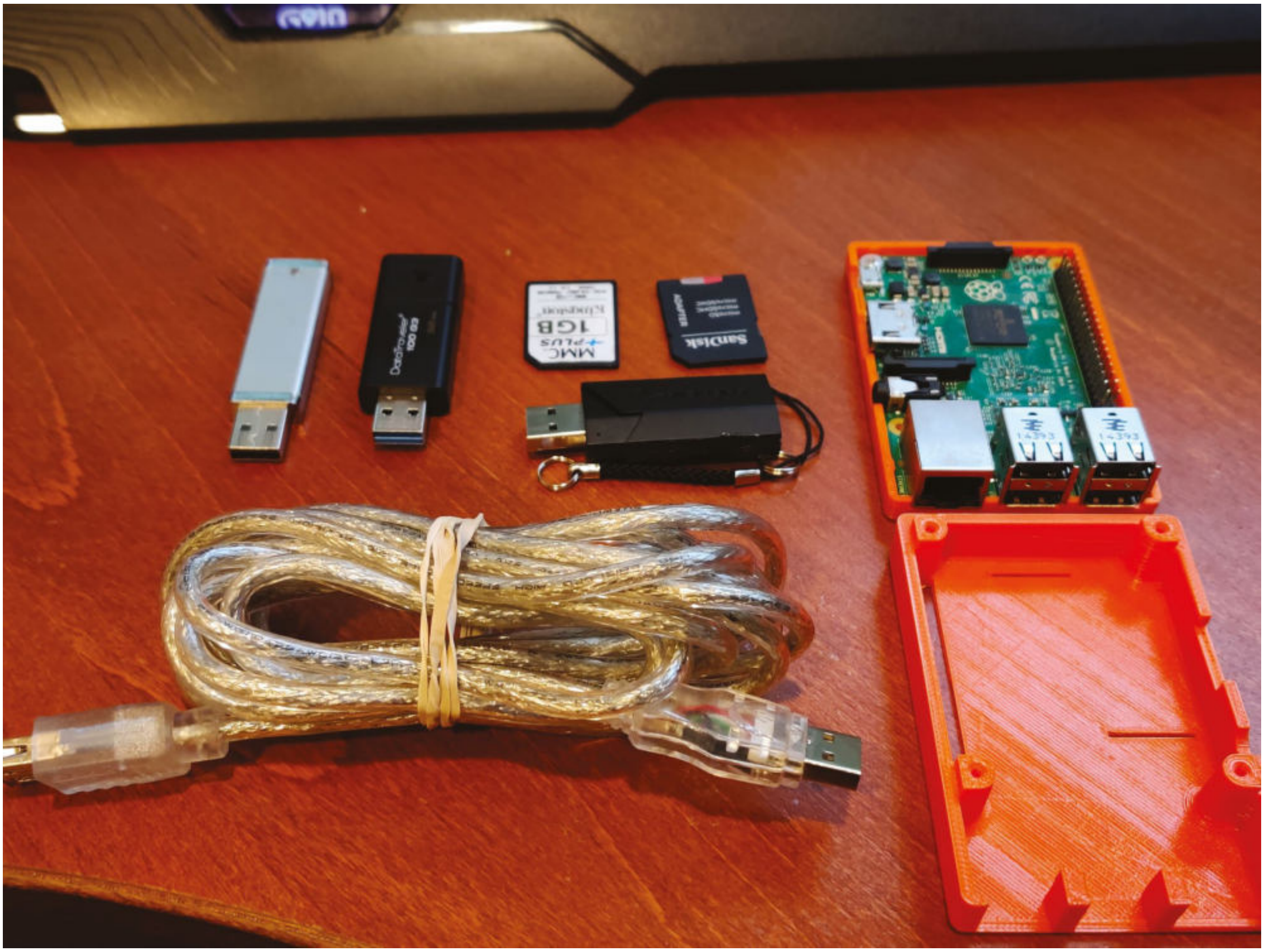
Need to read an SD card? Use the card reader

Quick FACTS

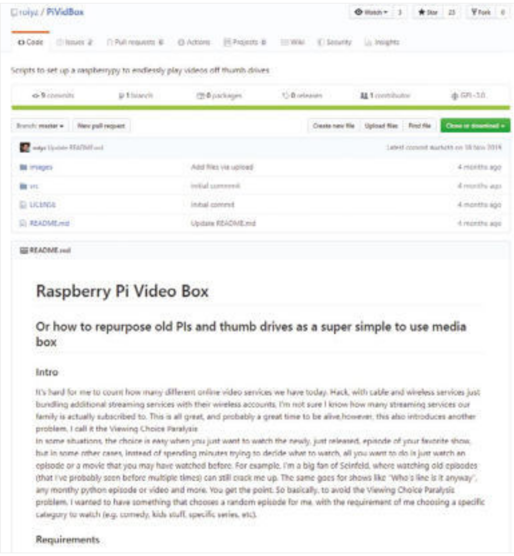
- PiVidBox can also take SD cards thanks to an adapter
- STL files are available to 3D-print the parts used
- It uses OMXPlayer to play videos instead of dedicated HTPC software
- Labels on the USB sticks and cards are simple images
- All the code is available on GitHub: **magpi.cc/pividbox**



The design for the USB and SD card holder; download the 3D print files from **magpi.cc/usbholder**



▲ The list of components is fairly simple – the only slightly quirky part is a USB A male to USB A female cable



roiyz / PiVidBox

Scripts to set up a raspberrypy to endlessly play videos off thumb drives

9 commits · 1 branch · 0 packages · 0 releases · 1 contributor · GPL-3.0

roiyz Update README.md		Latest commit … on 18 Nov 2019
images	Add files via upload	4 months ago
src	initial comment	4 months ago
LICENSE	Initial commit	4 months ago
README.md	Update README.md	4 months ago

README.md

Raspberry Pi Video Box

Or how to repurpose old PIs and thumb drives as a super simple to use media box

Intro

It's hard for me to count how many different online video services we have today. Hack, with cable and wireless services just bundling additional streaming services with their wireless accounts, I'm not sure I know how many streaming services our family is actually subscribed to. This is all great, and probably a great time to be alive,however, this also introduces another problem, I call it the Viewing Choice Paralysis.

In some situations, the choice is easy when you just want to watch the newly, just released, episode of your favorite show, but in some other cases, instead of spending minutes trying to decide what to watch, all you want to do is just watch an episode or a movie that you may have watched before. For example, I'm a big fan of Seinfeld, where watching old episodes (that I've probably seen before multiple times) can still crack me up. The same goes for shows like "Who's line is it anyway", any monthy python episode or video and more. You get the point. So basically, to avoid the Viewing Choice Paralysis problem, I wanted to have something that chooses a random episode for me, with the requirement of me choosing a specific category to watch (e.g. comedy, kids stuff, specific series, etc).

Requirements

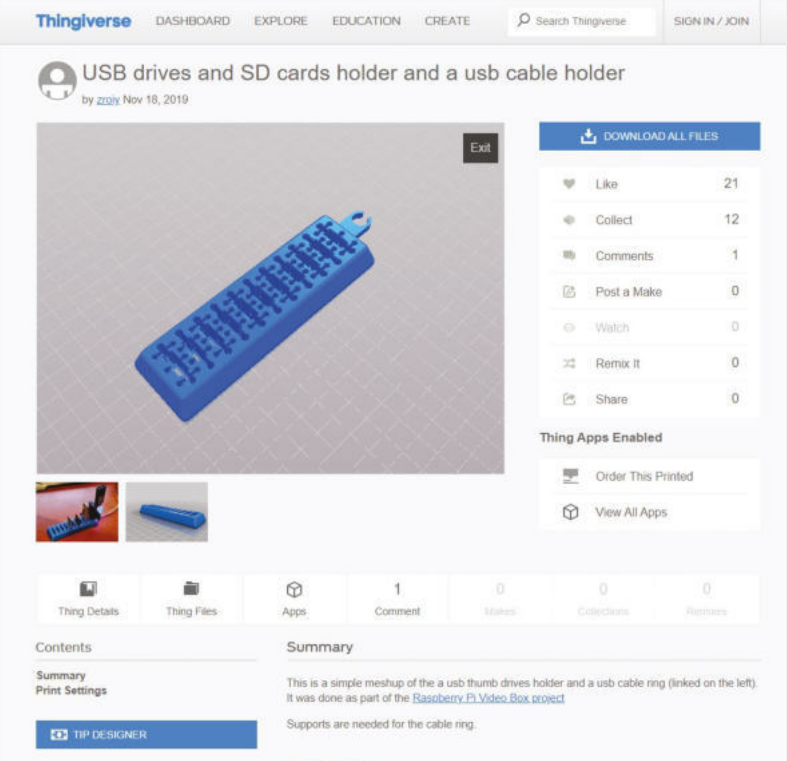
► All the code is available on the GitHub, including build instructions

> I had the aha moment where I was cleaning up my desk

pile of old Raspberry Pi [boards] and thumb drives that I wanted to use for something beneficial.

"I had the aha moment where I was cleaning up my desk, thinking of what to do with a my old Model 1 Raspberry Pi and a bunch of old small-capacity thumb drives (around 1GB to 2GB) when I heard my youngest son calling me to help him with some Baby Shark videos he really wanted to watch."

The system has been a success, with Roiy telling us that his kids use it without any problems. It's so good in fact, he's started using it more himself when they go to sleep.



Thingiverse DASHBOARD EXPLORE EDUCATION CREATE Search Thingiverse SIGN IN / JOIN

USB drives and SD cards holder and a usb cable holder

by zroiy Nov 18, 2019

Exit

DOWNLOAD ALL FILES

Like	21
Collect	12
Comments	1
Post a Make	0
Watch	0
Remix It	0
Share	0

Thing Apps Enabled

Order This Printed

View All Apps

Thing Details	Thing Files	Apps	1 Comment	0 Makes	0 Collections	0 Remixes

Contents

Summary
Print Settings

TIP DESIGNER

Summary

This is a simple meshup of the a usb thumb drives holder and a usb cable ring (linked on the left).
It was done as part of the Raspberry Pi Video Box project

Supports are needed for the cable ring.

▲ The 3D-printable files for the USB and SD card rack for this project are available on Thingiverse

Future plans

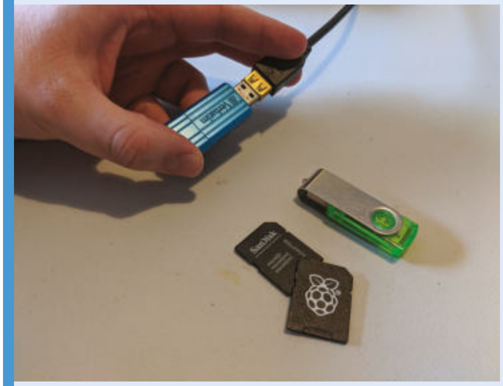
As with a lot of projects, this is not the end of the tinkering and tweaking.

"I have dozens of ideas on how to improve this," says Roiy. "For example: detecting if the HDMI connection is enabled and resume or pause based on the status. In this case, it pauses and unpauses the video as you switch the TV inputs. Support for extended media formats such as MP3 (so it can also be used to play music), adding Alexa or Google Home support so that skipping to the next video can be enabled by voice commands, and many other ideas."

We might have to put together a version ourselves for smaller relatives – and it's easy to do with the instructions on the PiVidBox GitHub page: **magpi.cc/pividbox**.

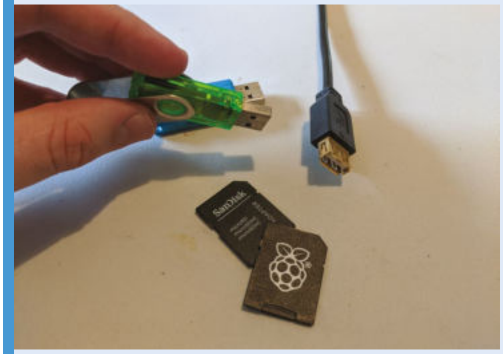
◀ Old USB sticks can easily be recycled in fun ways for more storage options

Playing a video



01 Select a USB stick and plug it into the PiVidBox. As it boots, it detects the USB drive that has been inserted.

02 A video is randomly selected and played, followed by a second video, on loop, forever. Great for kids.



03 If you want to change genre, just remove the USB stick and insert another one. PiVidBox picks up on the change and begins playing anew.

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

WIN! ARGON ONE THERMAL CASES
The MagPi
The official Raspberry Pi magazine
Issue 92
April 2020
magpi.cc
Track local weather & pollution
RASPBERRY PI
PROBLEMS SOLVED!
QUICK FIXES FOR TRICKY FAULTS!
Tweet with a hacked Morse Code Key
Make a Safe Cracker game
Solar powered security camera
43 PAGES OF PROJECTS & TUTORIALS

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

Rolling Monthly Subscription

- Low monthly cost (from £5)
- Cancel at any time
- Free delivery to your door
- Available worldwide

Subscribe for 12 Months

£55 (UK)	£90 (USA)
£80 (EU)	£90 (Rest of World)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero W Kit with Rolling Monthly Subscription)

☎ Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

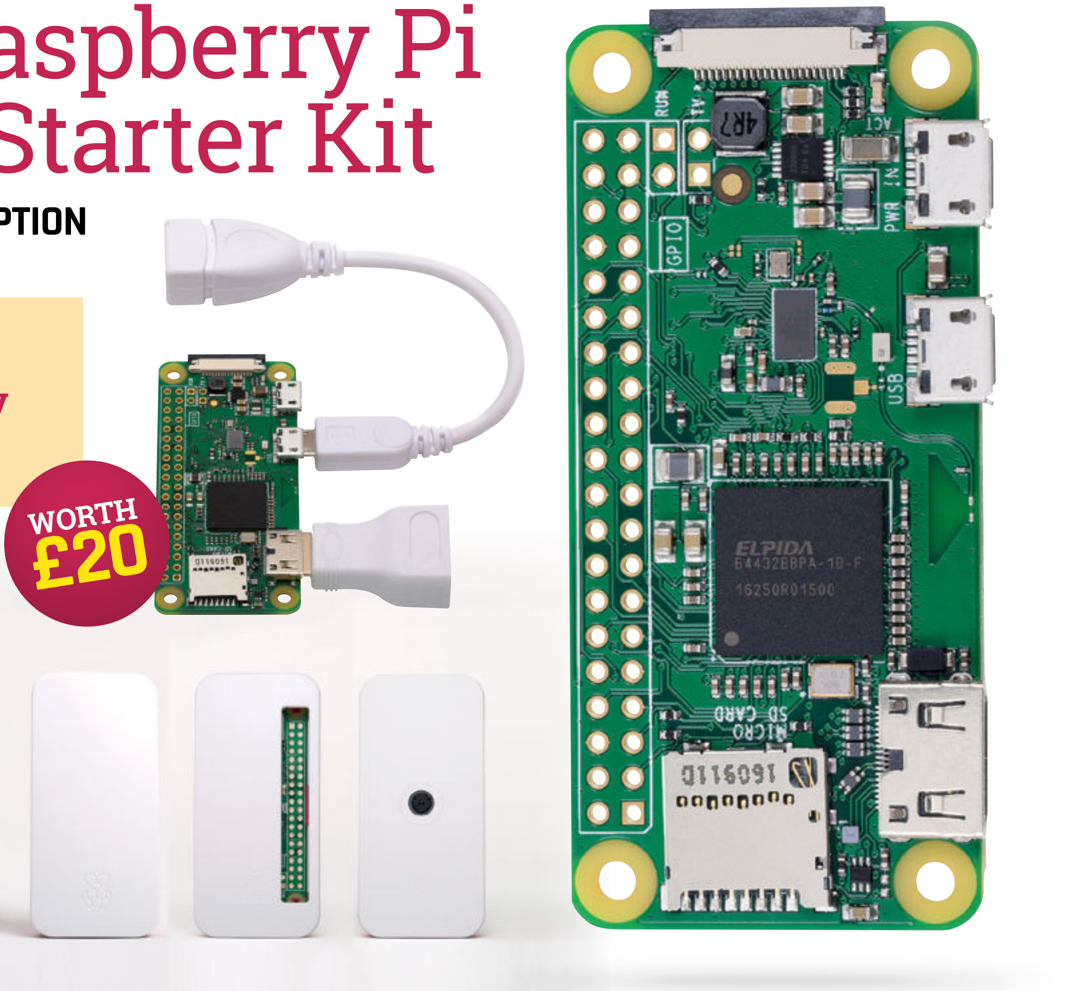
FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

Subscribe in print for 12 months today and you'll receive:

- Raspberry Pi Zero W
- Raspberry Pi Zero W case with three covers
- USB and HDMI converter cables
- Camera Module connector

Offer subject to change or withdrawal at any time



WORTH £20

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
#MonthOfMaking
Let's build together!
40 PAGES OF PROJECTS & TUTORIALS

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

RASPBERRY PI PROBLEMS SOLVED!

DIAGNOSE AND FIX ISSUES ON YOUR RASPBERRY PI WITH **ROB ZWETSLOOT**

Raspberry Pi is an incredible computer, and we love using ours to experiment like some sort of eighties movie LED and robot scientist. However, like everything, it's not flawless. Same with us Raspberry Pi users. Due to the unpredictable nature of mixing machine and human nature, sometimes things go a bit wrong.

This is where this article comes in. Everyone needs to learn how to correctly solve their computing issues, and through this easy-to-follow guide we'll show you some common fixes, starting from the basics, all the way up to more advanced issues. Put away the soldering iron and get ready to read.

SETTING UP YOUR FIRST RASPBERRY PI?

CHECK OUT OUR QUICKSTART GUIDE FOR COMPLETE STEP-BY-STEP INSTRUCTIONS

magpi.cc/quickstart

CONTENTS

FIRST TESTS

ALWAYS CHECK THESE BEFORE GOING ANY FURTHER...

01 Cables

Make sure that everything is properly plugged in to your Raspberry Pi – this includes power, HDMI, USB, Ethernet, audio, etc. If in doubt, give it a gentle jiggle in the slot and firmly push it in. Double-check the other end of the cables as well.

02 microSD card

Make sure a microSD card is inserted properly into the underside of your Raspberry Pi. Again, push it firmly in – if needed, take your Raspberry Pi out of its case before doing this.

03 Monitor

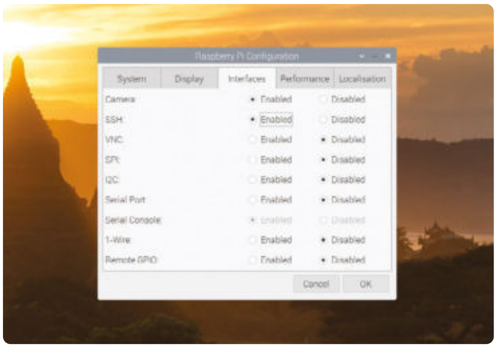
Connect to a monitor or display and turn it on before booting your Raspberry Pi – sometimes a Raspberry Pi won't output a video signal if there's nowhere for it to output to.

TOP 10 SIMPLE SOLUTIONS FOR SIMPLE PROBLEMS

EVERYONE'S A NEWBIE AT SOME POINT. HERE ARE SOME COMMON PROBLEMS YOU MAY NEED TO LEARN ABOUT

Lightning bolt / not enough power

A telltale sign of a Raspberry Pi not having enough power is a lightning bolt symbol appearing in the corner of the screen, or a peripheral/accessory not working properly. You need to make sure you have a powerful enough supply for your Raspberry Pi – this means a rating of 5 V 2.5 A for a Raspberry Pi 3B+ and below, or 5 V 3 A for a Raspberry Pi 4.



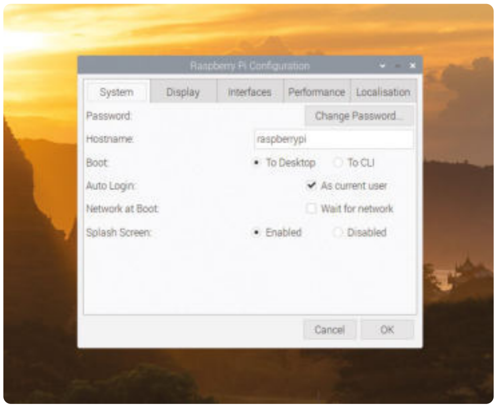
Raspberry Pi Configuration

System | Display | Interfaces | Performance | Localisation

Camera: ● Enabled ○ Disabled
SSH: ● Enabled ○ Disabled
VNC: ○ Enabled ● Disabled
SPI: ○ Enabled ● Disabled
I2C: ○ Enabled ● Disabled
Serial Port: ○ Enabled ● Disabled
Serial Console: ● Enabled ○ Disabled
1-Wire: ○ Enabled ● Disabled
Remote GPIO: ○ Enabled ● Disabled

Camera not detected

Using a Raspberry Pi Camera Module is usually straightforward. However, if it won't respond to commands then it may likely be a couple of things. First of all, check it's plugged in the correct way. Only one side of the ribbon cable has silver connectors and they need to line up with the connectors in the Camera slot. Secondly, you may need to enable the Camera in the Raspberry Pi Configuration tool – under the Interfaces tab.



Raspberry Pi Configuration

System | Display | Interfaces | Performance | Localisation

Password: Change Password...
Hostname: raspberrypi
Boot: ● To Desktop ○ To CLI
Auto Login: ☑ As current user
Network at Boot: □ Wait for network
Splash Screen: ● Enabled ○ Disabled

Desktop does not appear

If you've turned on your Raspberry Pi and are greeted with a command line instead of the desktop, it may have been told to boot to the command line instead. Run the command `startx` to boot into the desktop, then go to Raspberry Pi Configuration to change the option to boot 'To Desktop' – under the System tab.

Reusing a microSD card

Once upon a time, you were pretty easily able to transfer a working microSD card from one Raspberry Pi to another. Unfortunately, that doesn't always work any more, and you'll be forced to reinstall Raspbian (or your preferred OS) onto the card – see page 6. Make sure to back up any important files to a USB stick or the cloud before doing so, though – it will completely wipe the card.

General online help

If you have an issue that isn't covered here, then the best place to go is the Raspberry Pi Forum: **rpf.io/forums**. It's full of amazing and friendly people who have helped folks with many Raspberry Pi problems. Search for your problem first; then, if you can't find it, start a thread in the Troubleshooting section.

No picture on Raspberry Pi 4

We've all stumbled into this problem and spent far too long trying to fix it: long story short, Raspberry Pi 4 likes to boot from the 'HDMI 0' output, which is the one closest to the power port. You may just need to switch over the cable from the other HDMI port.

Code won't run

You've written out your code or copied it from the magazine and it won't run. What to do? You should get an error message when this happens, and it will list the problem and where it happened. If you don't understand the error message, make sure to check the line it mentions. If that doesn't help you spot an error, put the error message into your search engine of choice.

Booting quick fix

Having problems booting? Before you head over the page, try formatting and then reinstalling Raspbian (or whatever OS you prefer) and try again. While uncommon, installation issues can occur. This is especially common if you've just installed an OS.

Networking quick fix

Wireless LAN having issues? Ethernet causing a fuss? Sometimes you just need to reboot your Raspberry Pi. If that doesn't work, head over to our 'network fixes' section on page 34.

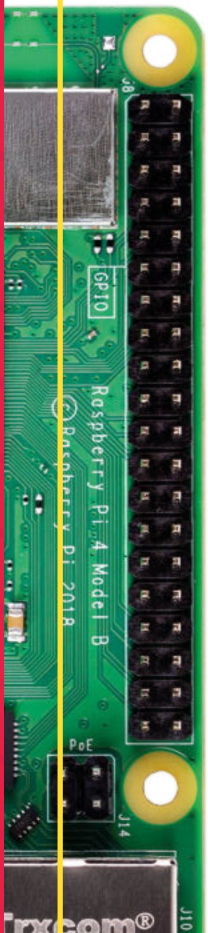
Software doesn't work

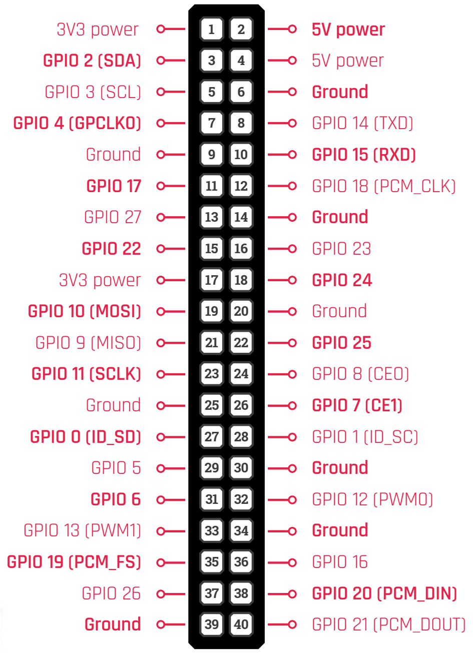
If a piece of software isn't running properly, then it may require an update. Open up the Terminal and type the following two lines to update all your software:

```
sudo apt update
sudo apt upgrade
```

GPIO pin doesn't work

If you're programming a button or LED or any other component connect to Raspberry Pi's GPIO, you need to make sure to reference the correct GPIO pin. Double-check your code and wire against the image below. If they're correct, you may just need to move it to a different available pin.





3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

SOLVING BOOT ISSUES

RASPBERRY PI NOT TURNING ON PROPERLY? HERE ARE SOME THINGS TO LOOK OUT FOR

General online help

Blinking ACT light

On the opposite end of the USB and Ethernet ports on a Raspberry Pi, you'll notice there are two little LEDs. The red one shows if there's power, while the green one is the ACT light. If the ACT light is blinking erratically during bootup, it means it's reading the microSD card fine. If it's not blinking at all, it means it can't read the microSD card and so you may need to use a new one or reinstall your OS – see page 6.

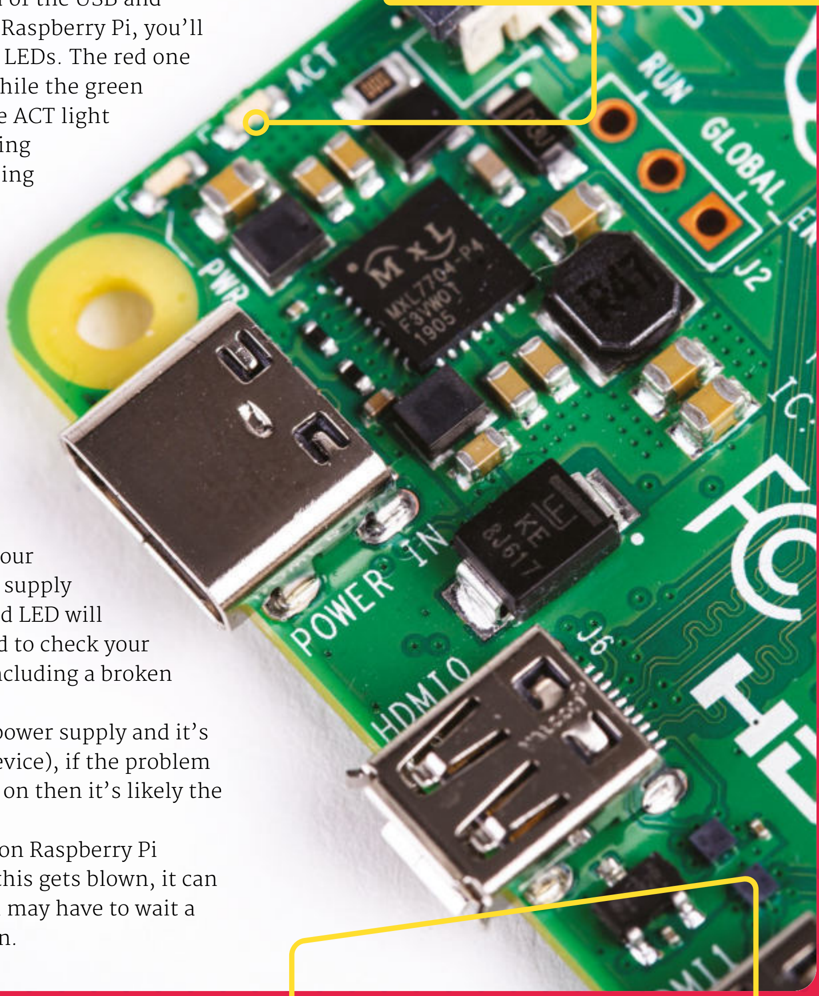
No power LED

Very simply, this means not enough or no power is getting to your Raspberry Pi. If the power supply drops below 4.65 V, this red LED will not light up. You may need to check your power supply for faults, including a broken connector, and replace it.

If you've checked your power supply and it's all working (on another device), if the problem Raspberry Pi isn't turning on then it's likely the board is broken.

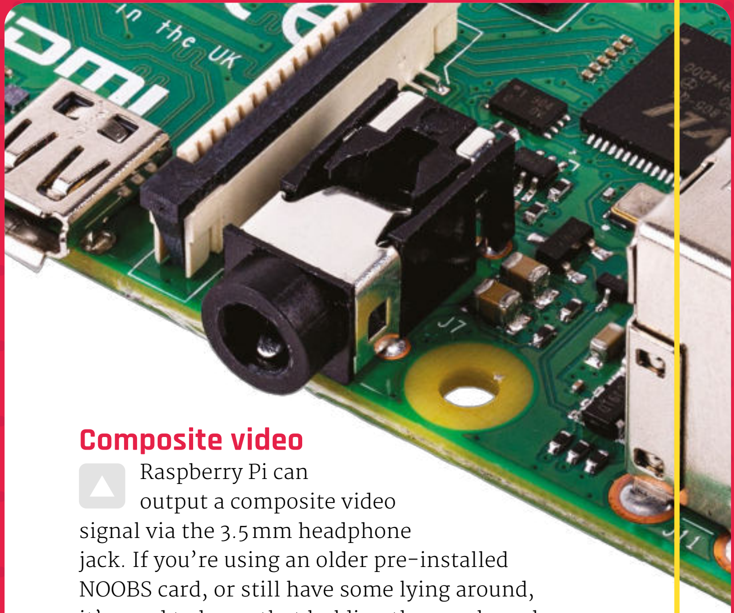
Note: the first-generation Raspberry Pi model has a polyfuse – if this gets blown, it can reset/repair itself, but you may have to wait a few days for this to happen.

THE ACT LED ALSO HAS SPECIFIC PATTERNS FOR SPECIFIC FAULTS. FIND OUT WHAT THEY ARE HERE:
magpi.cc/actled



Boot documentation

Need more advanced boot solutions? Check the documentation for Raspberry Pi on the Raspberry Pi website here: **magpi.cc/bootdocs**



Composite video

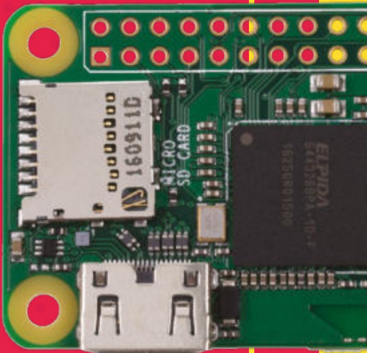
Raspberry Pi can output a composite video signal via the 3.5mm headphone jack. If you're using an older pre-installed NOOBS card, or still have some lying around, it's good to know that holding the number **3** key on a connected keyboard will force it to output via the composite output if that's what you're using. If you're in the US or another NTSC country, hold the **4** key instead.

Corrupt microSD card

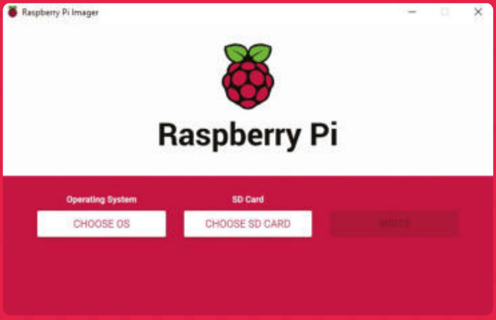
While uncommon, microSD cards can sometimes become corrupted while being used in a Raspberry Pi. Unfortunately, there's no solution to this other than getting a new microSD card, so make sure to back up your card regularly. You can do this by saving any important files, or even by making an image from your current card. We have a video on how to do this here: **magpi.cc/backupvid**

Raspberry Pi Zero

Power problems



Raspberry Pi Zero only has an ACT LED, so you will need to check your power supply if it won't boot up. We suggest trying the power and microSD card on another Raspberry Pi when in doubt.

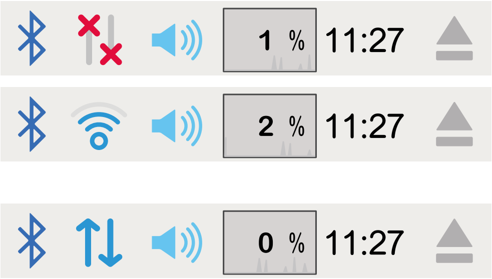


NOOBS issues

Not all versions of NOOBS work correctly with all Raspberry Pi Zero models. Instead of relying on NOOBS for your Raspberry Pi Zero, you should use the new Raspberry Pi Imager software to install Raspbian to a microSD card. You can find it at **magpi.cc/imager**. See page 6 of this issue for more details.

NETWORK FIXES

CAN'T CONNECT TO YOUR LAN? NO INTERNET? HERE ARE SOME THINGS YOU CAN DO



Understanding network symbols

The network symbols in Raspbian are pretty easy to read, and there are three with which you should be familiar. The first symbol is two grey lines with a red X on opposite ends – this denotes no network connection of any kind. Two blue arrows facing in opposite directions means you have a wired connection working, while the circle with waves emanating from it means you have a wireless connection.

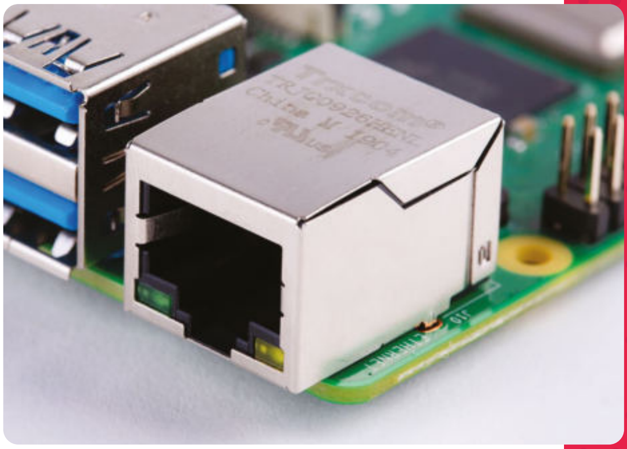
Speed differences

Different Raspberry Pi models have different connection speeds, especially on Ethernet connections. Raspberry Pi 3 and below max out at about 100Mbps as they use a 100MB adapter over USB. Raspberry Pi 3B+ is roughly 280Mbps, as it has a Gigabit Ethernet connection on a USB 2 line. Raspberry Pi 4, however, is a full gigabit. If you're having speed issues, make sure your router or switch supports up to the speeds you're expecting, and test with different Ethernet cables.

Desktop does not appear

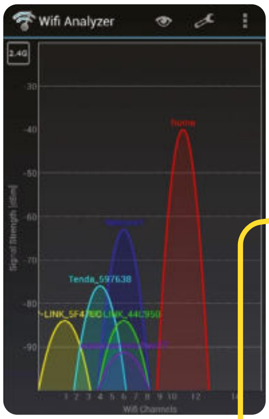
If you don't have a wired connection handy, you can add settings for your wireless network onto the boot folder of a Raspberry Pi microSD card. Create a file there called **wpa_supplicant.conf** and add your network details like so:

```
 ctrl_interface=DIR=/var/run/wpa_
supplicant GROUP=netdev
 update_config=1
 country=<Insert country code here>

 network={
  ssid="<Name of your WiFi>"
  psk="<Password for your WiFi>"
 }
```



Wireless testing

There are a lot of things that can interfere with your wireless reception in your home. Distance from your router, interference from neighbours, walls, and even just too many people on the same channel. We like to use apps like Wifi Analyzer to test locations around our home to find out the best spots to place wireless devices such as a Raspberry Pi.



SSH connection

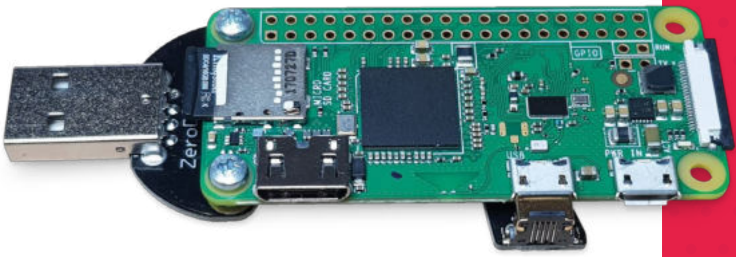
Many security precautions have been implemented in the last couple of years to make sure Raspbian and Raspberry Pi are more secure – this includes making sure people change the password on their Raspberry Pi, and keeping SSH off. You can turn it on from the Interfaces tab in Raspberry Pi Configuration – or, if you're using a headless Raspberry Pi, simple place an empty file called **ssh** in the boot folder of the microSD card.

Wired Raspberry Pi Zero

Raspberry Pi Zero does not include an Ethernet port. However, if you need to use a wired connection, you can always use a USB Ethernet dongle. It may not be as speedy as other Raspberry Pi models, but it will do the job.

Raspberry Pi Zero and A gadget issues

A workaround for a lot of people wanting a portable, network connected computer is to put it into 'gadget' or USB boot mode. If you're having trouble and want to deactivate it, either flash the microSD card (see page 6) or head to the usbboot documentation to fix your setup: **magpi.cc/usbboot**



Networking documentation

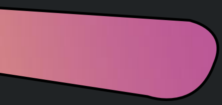
Not finding your problem here? Check the configuration documentation for more info on networking with Raspbian and Raspberry Pi: **magpi.cc/configdocs**

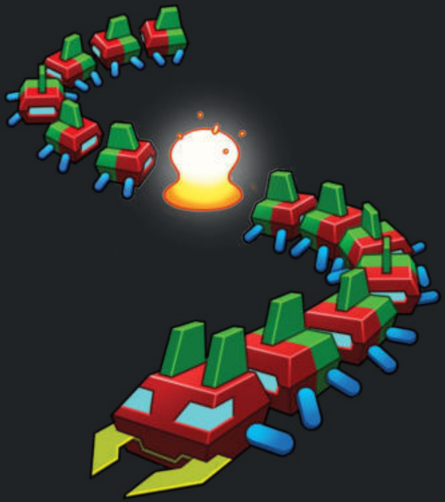
Networking booting and PoE

Raspberry Pi 3/3B+ and 4 support network booting, and Raspberry Pi 3B+ and 4 even support Power over Ethernet with the PoE HAT.

These are advanced methods and have some issues you might come across, such as DHCP timing out, ARP check failing, and others. For full details on how to set it up and troubleshoot, we suggest heading to the network booting documentation here: **magpi.cc/netboot**

CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²
Raspberry Pi PRESS





Part 02

Getting started with electronics: Sensors

One of the great things about Raspberry Pi is that it has GPIO pins to attach your own electronic circuits

MAKER

Simon Monk

Simon divides his time between writing and designing products for MonkMakes Ltd. Some of his better-known books include *Programming Raspberry Pi* (TAB) and *The Raspberry Pi Cookbook* (O'Reilly).

simonmonk.org

You'll Need

- The Mu Python editor **codewith.mu**
- Solderless breadboard
- 3 × Female-to-male jumper wires
- 2 × 1kΩ resistors
- NTC thermistor
- NPN phototransistor (visible light wavelength)
- 330 nF capacitor

In this tutorial, you will learn how to sense both temperature and light, using your Raspberry Pi and a few simple components. One project will allow you to measure temperature using a component called a thermistor, and the other project acts as a rudimentary light meter. Both display their readings in a window on your Raspberry Pi's screen using the guizero Python library (**magpi.cc/guizero**).

Both projects also make use of the PiAnalog (**magpi.cc/pianalog**) Python library that lets you connect analogue sensors to Raspberry Pi without special hardware.

Although these projects sense temperature and light, you could easily adapt them to use other types or resistive sensor, including stress sensors, variable resistors, and even some types of gas sensor.

The thermometer project

01 Install the code

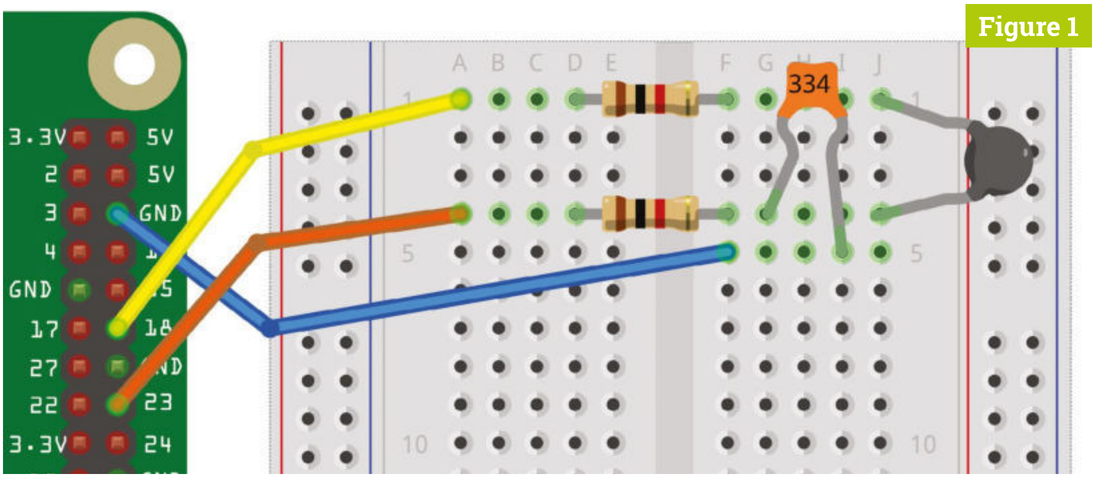
Before fetching the code from the internet, you should run Mu, which you will find in the Programming section of your main menu. If it's not there, update your system to the latest version of Raspbian (**magpi.cc/raspbianupdate**).

Running Mu ensures that the **mu_code** directory is created, into which we will now copy the program code. To do this, open a Terminal window and run the commands:

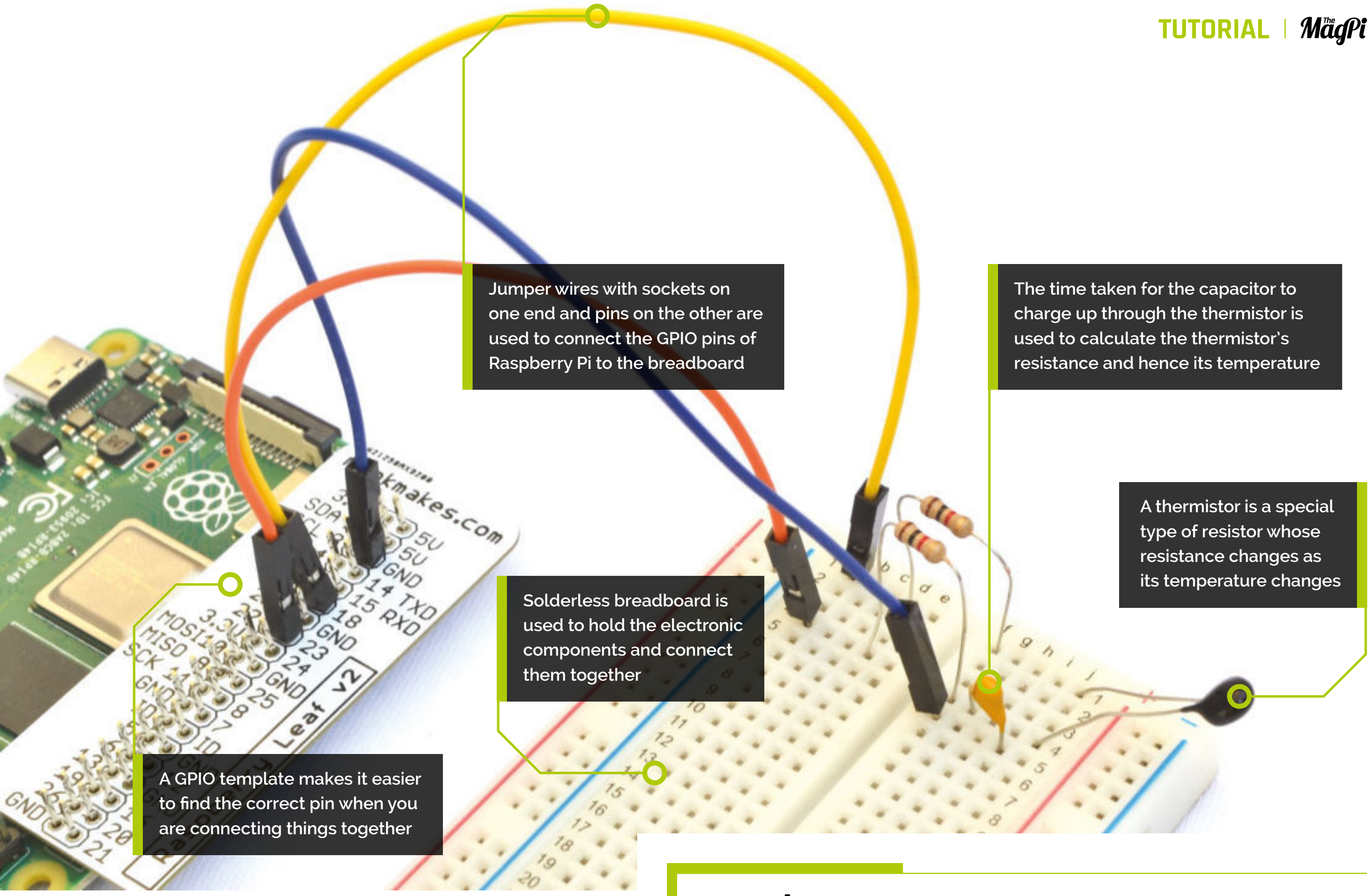
```
wget http://monkmakes.com/downloads/pb1.sh
sh pb1.sh
```

This will copy the programs used in this tutorial into the **mu_code** directory, along with some others.

Figure 1



▲ **Figure 1** The thermometer wiring diagram



Jumper wires with sockets on one end and pins on the other are used to connect the GPIO pins of Raspberry Pi to the breadboard

The time taken for the capacitor to charge up through the thermistor is used to calculate the thermistor's resistance and hence its temperature

A thermistor is a special type of resistor whose resistance changes as its temperature changes

Solderless breadboard is used to hold the electronic components and connect them together

A GPIO template makes it easier to find the correct pin when you are connecting things together

> In this project, none of the components needs to be a particular way around

02 Place components onto breadboard

Using **Figure 1** as a reference, push the jumper wires into the breadboard at the positions shown. Bend the resistor legs so that they fit into the holes.

The five holes in each row on the breadboard are connected together under the plastic, so it's very important to get the correct row for your component leg. In this project, none of the components needs to be a particular way around.

03 Connect breadboard to Raspberry Pi

Again, using **Figure 1** as a reference, connect the GPIO pins on Raspberry Pi to the breadboard. A GPIO template will make this easier – if you don't have one, you'll need to carefully count the pin positions. It doesn't matter what colour jumper wires you use, but if you stick to the colours used in the diagram, it's easier to check that your wiring is correct.

05_thermometer.py

Language: **Python 3**

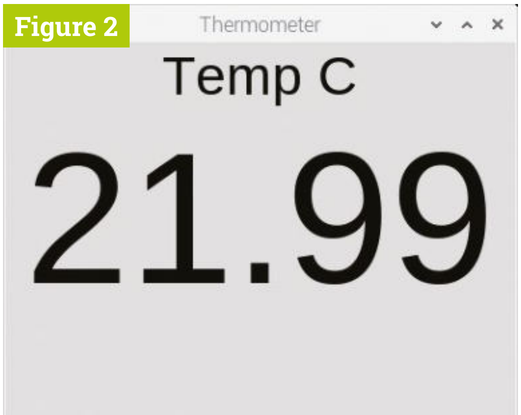
```python
# 05_thermometer.py
# From the code for the Box 1 kit for the Raspberry Pi by MonkMakes.com

from PiAnalog import *
from guizero import App, Text
import time

p = PiAnalog()

# Update the temperature reading
def update_temp():
    temperature = p.read_temp_c()
    temperature = "%.2f" % temperature # Round the temperature to 2 d.p.
    temp_text.value = temperature
    temp_text.after(1000, update_temp)

# Create the GUI
app = App(title = "Thermometer", width="400", height="300")
Text(app, text="Temp C", size=32)
temp_text = Text(app, text="0.00", size=110)
temp_text.after(1000, update_temp) # Used to update the temperature reading
app.display()
```

Top Tip

Using a photoresistor

You can also use a 1 kΩ photoresistor (sometimes called an LDR – light dependent resistor) in place of the phototransistor. Photoresistors are becoming less common in projects as they contain cadmium, an element whose use is restricted in the RoHS (Restriction of Hazardous Substances) legislation. Don't worry: a photoresistor is not dangerous to touch, as it's all sealed up in plastic.

Figure 2



Thermometer

Temp C

21.99

▲ **Figure 2** Displaying the temperature using guizero

04 Running the program

Load and run the **05_thermometer.py** program using Mu. The code is configured for the thermistor supplied with the MonkMakes Project Box 1 kit. If you look near the top of the file, you will see the line: `temperature = p.read_temp_c()`.

If you are using your own thermistor, you will need to add two new parameters to the method call. The first parameter is the value of Beta for the thermistor, and the second the value to R25 (resistance at 25°C). You will find both of these values in the thermistor's datasheet. For example, if Beta is 3800 and R25 is 1 kΩ, you would use: `p.read_temp_c(3800, 1000)`.

After a few seconds, a window will appear, like the one in **Figure 2**, displaying the temperature. If you would rather have the temperature displayed in degrees Fahrenheit, then run the program **05_thermometer_f.py** instead.

> “If you are using your own thermistor, you will need to add two new parameters to the method call”

05 Changing the temperature

The easiest way to change the temperature of the thermistor is to pinch it between your fingers so that your body warmth heats it up. You should see the temperature steadily increase and then decrease back to room temperature when you let go of the thermistor.

Light meter project

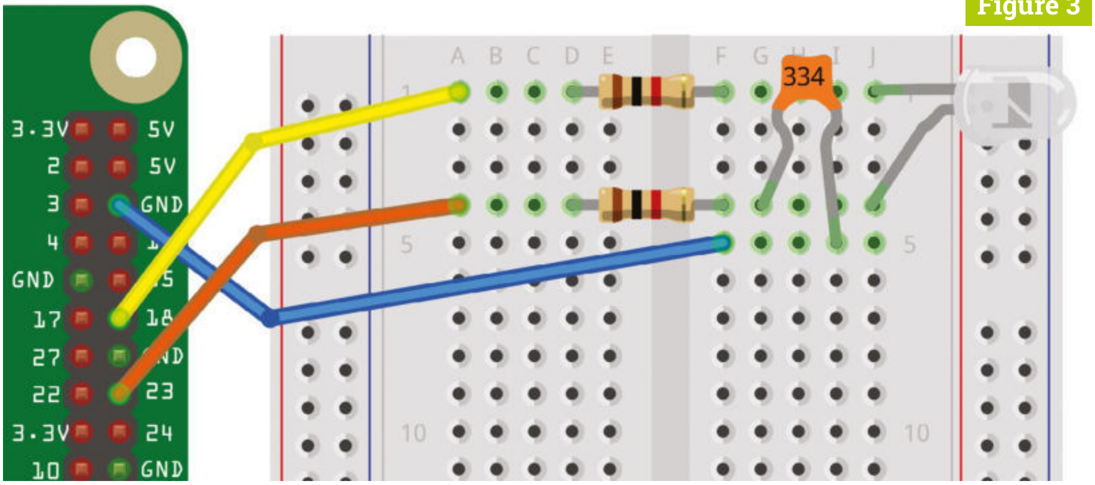
01 Disconnect the breadboard

This project has almost the same layout as the thermometer, and we are just going to swap the thermistor for a phototransistor, but it is still a good idea to disconnect the breadboard from your Raspberry Pi. First, pull the jumper wires off the GPIO pins on Raspberry Pi and then take the thermistor off the breadboard.

02 Place phototransistor onto breadboard

This time, using **Figure 3** as a guide, put the phototransistor legs into the breadboard. The

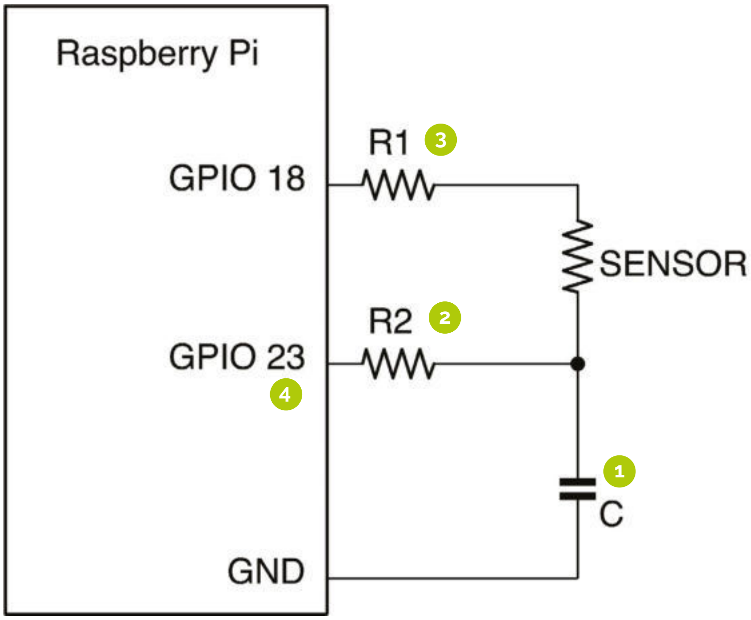
Figure 3



▶ **Figure 3** The light meter wiring diagram

A schematic diagram for using resistive sensors with Raspberry Pi

1. As a capacitor fills with charge, its voltage increases; when this reaches about 1.6 V, it will appear as HIGH on Raspberry Pi's GPIO pin acting as a digital input. If under 1.6 V, it will appear as LOW on the digital input
2. Before we can use the capacitor (C) to measure the resistance of the sensor, it first needs to be discharged (emptied). To do this, GPIO 18 is set to be an input (so almost no current flows through it) and GPIO 23 is set to a digital output at 0 V. This discharges C through resistor R2
3. Having used GPIO 23 and R2 to discharge the capacitor, GPIO 23 is set to be an input (to monitor the voltage of the capacitor) and GPIO 18 is now set to be a digital output at 3.3 V. The capacitor (C) will now start to charge and the voltage across it will steadily increase at a rate that depends on the resistance of R1 (fixed) and the sensor (varying with temperature/light/etc. depending on the sensor type)
4. Once the voltage across the capacitor and therefore at GPIO 23 reaches about 1.6 V, the input is read as high by the monitoring Python program, and the time taken for this to happen is used to calculate the resistance of the sensor



phototransistor must go the correct way around: the longer leg should go to row 4.

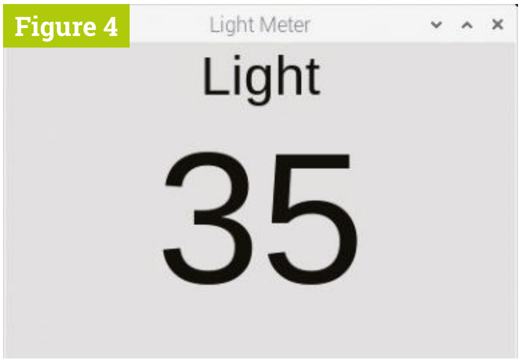
03 Connect breadboard to Raspberry Pi

Using **Figure 3** as a reference, connect the GPIO pins on Raspberry Pi to the breadboard using three female-to-male jumper wires.

04 Running the program

To use the light meter, load and run the program **08_light_meter.py** in Mu. When the program starts, a window like that in **Figure 4** will appear, showing the light level. Try shading the phototransistor with your hand or shining a light on it to see how the readings change.

Figure 4



Light

35

Figure 4 Displaying the light level

08_light_meter.py

Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/pibox1git

```python
# 08_light_meter.py
# From the code for the Box 1 kit for the Raspberry Pi by MonkMakes.com

from guizero import App, Text
from PiAnalog import *
import time, math

p = PiAnalog()

def light_from_r(R):
    # Log the reading to compress the range
    return math.log(1000000.0/R) * 10.0

# group together all of the GUI code
# Update the reading
def update_reading():
    light = light_from_r(p.read_resistance())
    reading_str = "{:.0f}".format(light)
    light_text.value = reading_str
    light_text.after(200, update_reading)

app = App(title="Light Meter", width="400", height="300")
Text(app, text="Light", size=32)
light_text = Text(app, text="0", size=110)
light_text.after(200, update_reading)
app.display()
```

Tweet with Morse code

If your keyboard has you bored, why not learn Morse and then be able to send tweets using nothing but a simple switch?

MAKER

PJ Evans

PJ is a writer and software engineer who seems to want to make communicating with the outside world as challenging as possible.

@mrpjevans

You'll Need

- Small breadboard **magpi.cc/minibreadboard**
- Tactile switch **magpi.cc/switches**
- Adafruit 16×2 LCD Keypad Kit (optional) **magpi.cc/keypadkit**
- Morse key (optional) **magpi.cc/morsekey**
- Twitter account (optional) **twitter.com**
- Jumper wires

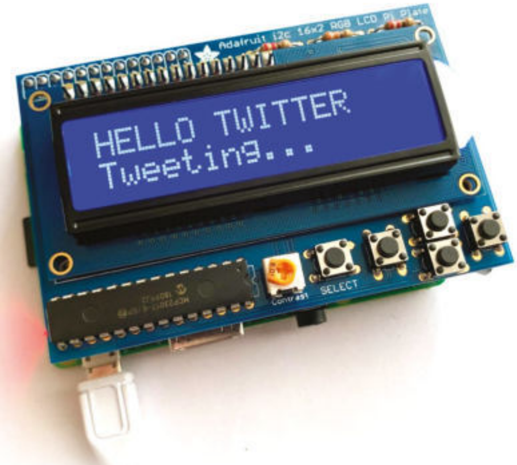
When Samuel Morse created his communication system in the mid-1800s, it revolutionised wireless communication. The original idea was that simple electronic pulses could be sent further and more reliably than voice, and so the Morse alphabet was used to describe each letter and number as a combination of short pulses (dots) and long pulses (dashes).

Learning Morse code is a challenge, but can be rewarding and a lot of fun. As an introduction to this classic way of communicating, we're going to build a tweeting Morse code key. We'll learn how to interpret input and also how you can create more complicated projects by breaking them down into small pieces.

01 Let's get set up

First, select the right Raspberry Pi model for the job. Of course, we would heartily recommend a Raspberry Pi 4, but in fact this project will not be too demanding on even the oldest models, so it is great for upcycling an older Raspberry Pi computer. In fact, it will even work with the original Model A and B.

Start by installing the latest version of Raspbian. We've no need for a graphical user interface, so you can use Raspbian Lite if you wish; whatever is most comfortable. We'll be doing everything in the command line.



This popular display means we can see our Morse code without the need for a monitor

02 Configure and update

This project has several steps, so don't worry if you just want to practise Morse code – we'll get to that first. If you want to complete everything here, you'll need to set up an internet connection (wireless or wired) and enable I²C, which is used to communicate with the LCD screen. By running `sudo raspi-config` from the command line, you can enable WiFi under 'Network Options' and I²C under 'Interfacing Options'.

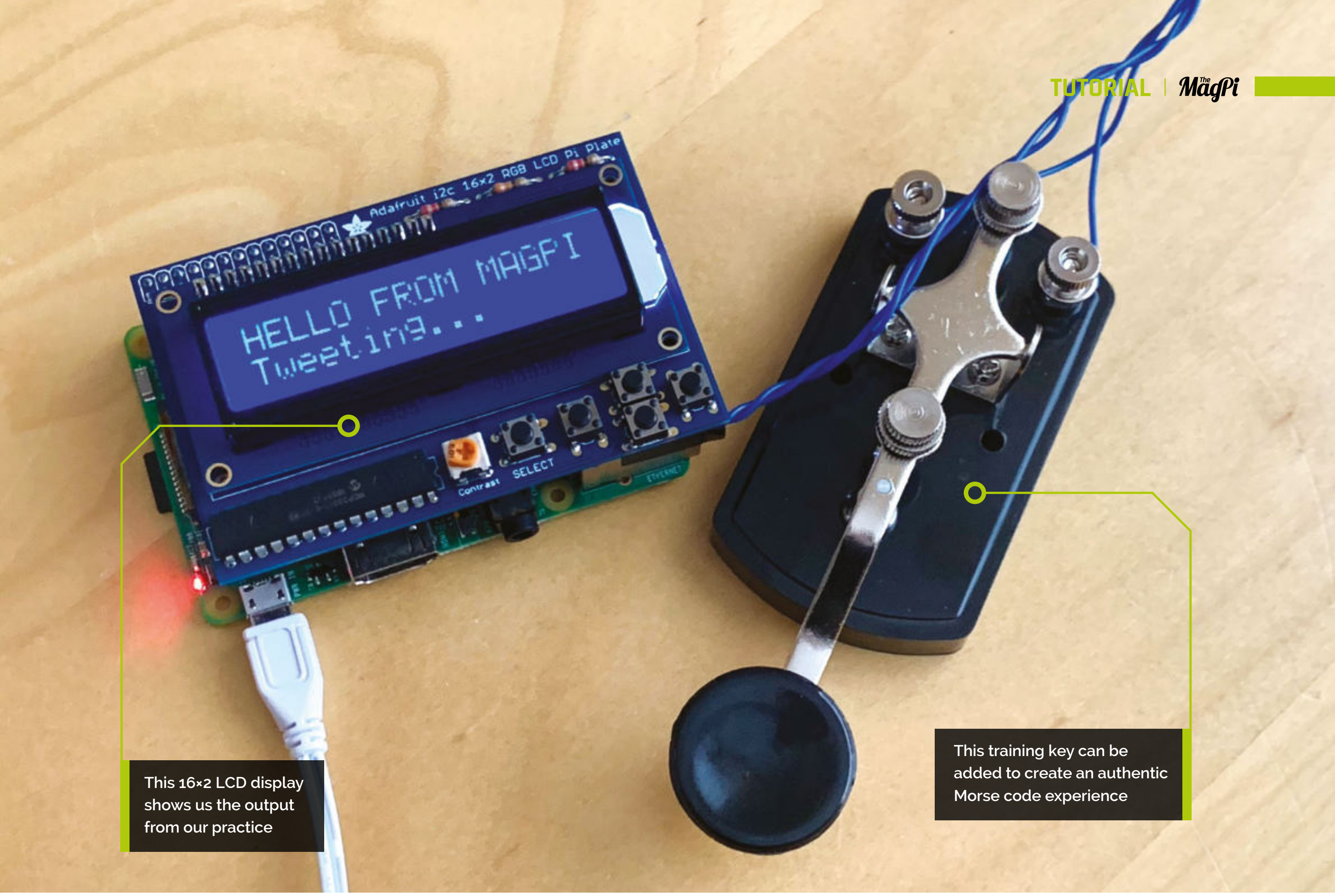
Whatever it is that you've decided to do, always make sure you've updated the system by running `sudo apt update && sudo apt upgrade`. This may take some time; once complete, it's important you reboot so that I²C is properly enabled.

03 Get switched on

Let's try to emulate the Morse key by using a tactile switch. These widely available and inexpensive switches make a satisfying 'click' when pressed (hence the name). They have four pins – two pairs that are connected on the longer side, so the switching is done between those with the shorter gap. Bearing this in mind, place the switch into the breadboard so the longer edge follows the connected rows. Don't worry if you make a mistake: nothing can be damaged. Now connect the breadboard to your Raspberry Pi's GPIO. Run jumper leads on each side of the switch to the last two pins at the end (nearest the USB ports) of the GPIO header: GND and GPIO 21.

04 Coding time

Once you've checked all your connections, create a new file called **morse.py** in your favourite editor and enter the code listed here. You can also download it from **magpi.cc/twitterkey** if you don't fancy the typing. The code will listen for changes to the button's 'state' (whether it is pressed or not) and measure the time differences



This 16×2 LCD display shows us the output from our practice

This training key can be added to create an authentic Morse code experience

to work out whether you made a 'dot' or a 'dash'. It will then convert the pattern into a letter and display it on the screen.

First, install these dependencies (libraries that help us):

```
sudo apt install python3-pip
pip3 install gpiozero
```

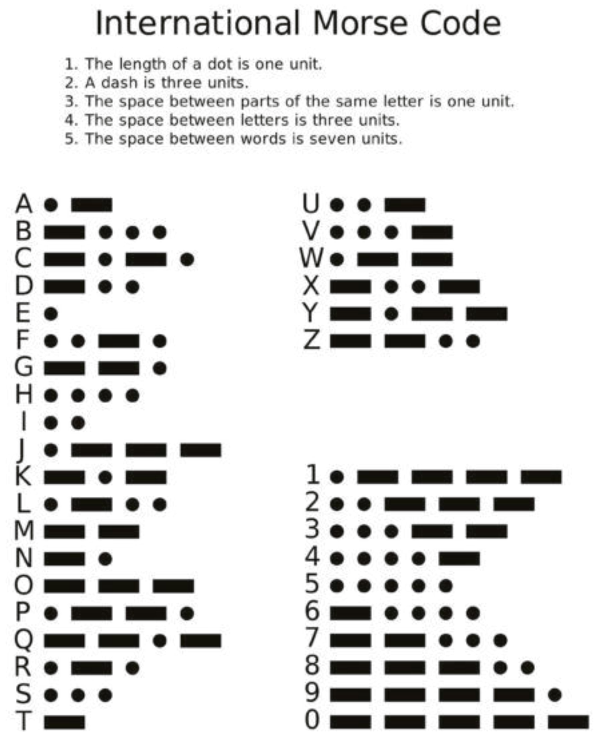
Then run `python3 morse.py`.

05 Practise dots and dashes

Using the chart printed here (**Figure 1**), see if you can spell your name out by clicking the button. Use a quick press for a 'dot' and a slightly longer press for a 'dash'. Leave the button untouched for a slightly longer time to tell the code you've finished your letter. Once you're happy everything is working and you've had some fun, **CTRL+C** will stop the program.

If you're not happy with the timings, you can adjust them to suit your 'fist' (the name operators give their style of keying). You can adjust the timings for a dot, dash, and interval between letters by changing the timings in the variables `dot_timeout` and `dash_timeout` at the start of the code. Don't be afraid to experiment.

Figure 1



International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● ▬
B	▬ ● ● ●
C	▬ ● ▬ ●
D	▬ ● ●
E	●
F	● ● ▬ ●
G	▬ ▬ ●
H	● ● ● ●
I	● ●
J	● ▬ ▬ ▬
K	▬ ● ▬
L	● ▬ ● ●
M	▬ ▬
N	▬ ●
O	▬ ▬ ▬
P	● ▬ ▬ ●
Q	▬ ▬ ● ▬
R	● ▬ ●
S	● ● ●
T	▬
U	● ● ▬
V	● ● ● ▬
W	● ▬ ▬
X	▬ ● ● ▬
Y	▬ ● ▬ ▬
Z	▬ ▬ ● ●
1	● ▬ ▬ ▬ ▬
2	● ● ▬ ▬ ▬
3	● ● ● ▬ ▬
4	● ● ● ● ▬
5	● ● ● ● ●
6	▬ ● ● ● ●
7	▬ ▬ ● ● ●
8	▬ ▬ ▬ ● ●
9	▬ ▬ ▬ ▬ ●
0	▬ ▬ ▬ ▬ ▬

▲ **Figure 1** The Morse code alphabet. Although it appears random, there is an underlying structure that helps you understand and memorise the patterns

Top Tip

Learning Morse

There are many resources online that will help you understand how Morse code is structured. Start with **learnmorsecode.info**

06 Build the LCD

To allow us to create Morse code without the need for a full display, we've selected a bright, crisp and slightly-retro LCD screen from Adafruit. These popular panels normally require a lot of GPIO pins to be driven natively, but this HAT uses an input extender so that only two pins are needed. Better yet, it comes with five tactile switches on-board so we can use them for input.

This LCD kit arrives unassembled, so it's time to get the soldering iron out. There are excellent assembly instructions at **magpi.cc/lcdkitinfo**. As always, read through them before doing anything and take your time.

07 Set up the display

Once your display is assembled, you can attach it to your Raspberry Pi (make sure the latter is switched off!). Due to the close proximity of some of the resistors, cover the top of the USB ports and Ethernet port with insulation tape if they are close to the PCB of the display.

Test the display is working by installing its Python libraries:

```
sudo pip3 install adafruit-circuitpython-charlcd
```

Now create a new file called **lcd.py** and enter the code from the listing here. Save it and run it using `python3 lcd.py`. The display should show your message. If it lights up but you can't see anything, adjust the 'Contrast' potentiometer until the text appears.

lcd.py

> Language: **Python 3**

```python
import board
import busio
import adafruit_character_lcd.character_lcd_rgb_i2c as character_lcd
lcd_columns = 16
lcd_rows = 2
i2c = busio.I2C(board.SCL, board.SDA)
lcd = character_lcd.Character_LCD_RGB_I2C(i2c, lcd_columns, lcd_rows)
lcd.color = [100, 0, 0]
lcd.message = "Hello\nFrom MagPi"
```

08 Version 2

It's time for a more advanced version of our original code. This one is a bit longer, so download **lcd_morse.py** from **magpi.cc/twitterkey**. This time we're reading input from the LCD's on-board tactile keys, so the code needs to be a bit different. The time measurement variables are still there. Run it using `python3 lcd_morse.py`.

You should be able to key away and see the interpreted letters appear on screen. You now have a functioning standalone Morse code trainer.

09 Let's tweet

We'd like to be able to send our messages to Twitter. For security reasons, we need to create a Twitter 'application' which gives the code unique credentials for posting on our behalf. We're using the python-twitter library – see **magpi.cc/pytwitterinfo** for an excellent tutorial on how to set it up. You will be given four strings: a consumer key, consumer secret, access token, and access token secret. Enter all the values in the equivalent variables in the first few lines of **lcd_morse_twitter.py** (download the code from **magpi.cc/twitterkey**). Now save the file.

10 Tweet with Morse!

Run `python3 lcd_morse_twitter.py`. As before, you can construct your message by tapping on the right-hand cursor button of the LCD display. Your message will be displayed at the top, and the current dots and dashes in the 'buffer' at the bottom. Made a mistake? No problem: click the left-hand cursor to delete the previous character or the 'up' key to delete the entire message and start again. When you're happy, click on 'Select' to send. Your message will be posted to your account for all of Twitter to read.

11 Add a Morse key

Let's take the authenticity up a notch by adding a real Morse key. These keys are nothing more than a simple on/off switch. That said, some can be surprisingly expensive as they are built using precision components to allow the operator to go faster and faster with fewer mistakes. We've selected a more affordable training key that has two contacts that can be directly connected. To use the

morse.py

▶ Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/twitterkey

```python
from gpiozero import Button
import time

button = Button(21)  # GPIO Pin 40
dot_timeout = 0.15
dash_timeout = 1
current_letter = ""

morse = {
    ".-": "A", "-...": "B", "-.-.": "C", "-..": "D", ".": "E",
    "..-.": "F", "--.": "G", "....": "H", "..": "I", ".---": "J",
    "-.-": "K", ".-..": "L", "--": "M", "-.": "N", "---": "O",
    ".--.": "P", "--.-": "Q", ".-.": "R", "...": "S", "-": "T",
    "..-": "U", "...-": "V", ".--": "W", "-..-": "X", "-.--": "Y",
    "--..": "Z", ".----": "1", "..---": "2", "...--": "3", "....-": "4",
    ".....": "5", "-....": "6", "--...": "7", "---..": "8", "----.": "9",
    "-----": "0"
}

while True:

    # Wait for a keypress or until a letter has been completed
    button.wait_for_press(dash_timeout)

    # If we've timed out and there's been previous keypresses, show the letter
    if button.is_pressed is False and len(current_letter) > 0:
        print("\nMorse: " + current_letter)
        if current_letter in morse:
            print("Letter: " + morse[current_letter])
        else:
            print("Not recognised")
        current_letter = ""

    elif button.is_pressed:

        # The key has been pressed, work out if it's a dot or a dash
        button_down_time = time.time()
        button.wait_for_release()
        button_up_time = time.time()
        button_down_length = button_up_time - button_down_time

        # Was it a dot or dash?
        if button_down_length > dot_timeout:
            print('-', end='', flush=True)
            current_letter += '-'
        else:
            print('.', end='', flush=True)
            current_letter += '.'

    time.sleep(0.1)
```

existing code, solder two wires to the underneath of the rightmost tactile switch on the LCD board and connect them to the key using its screw terminals. Now you can key away using the real thing!

12 Going further

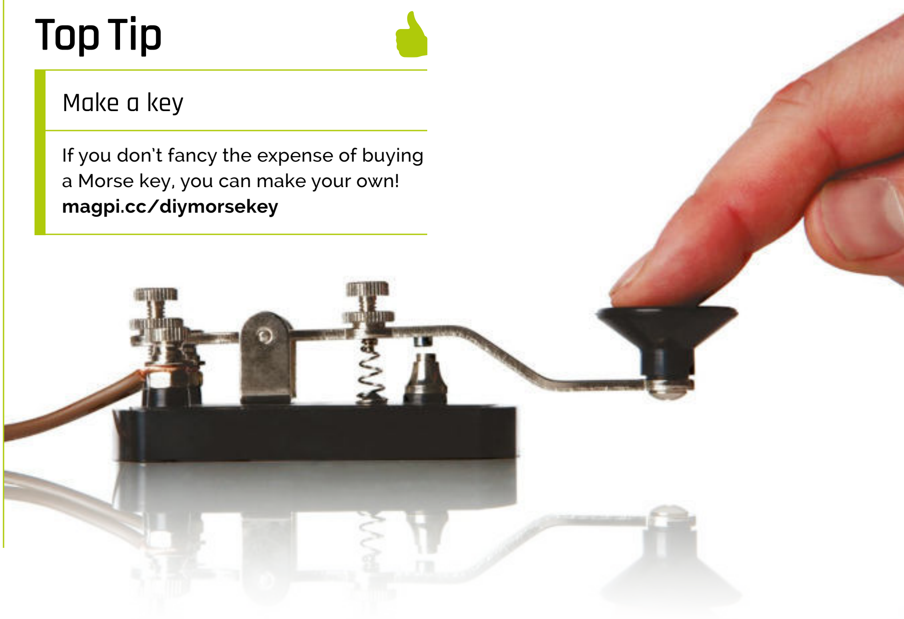
Now you have the basics as Python code, you can repurpose your tweeting Morse key for anything you can imagine. Add a second key and create Morse code challenge games. How about Morse code hangman? Add timing in to see how many letters per minute you can key. Could two identical setups send messages to each other?

Although initially challenging, learning Morse code is rewarding and can inspire operators to go on to the rich and fascinating world of amateur radio. Over to you.

Top Tip

Make a key

If you don't fancy the expense of buying a Morse key, you can make your own! **magpi.cc/diymorsekey**



Part 02

Use Swarm with Raspberry Shake seismographic data

Use your Raspberry Shake (or tap into the Shake community) and work with seismographic data using Swarm

MAKER

Lucy Hattersley

Lucy is editor of The MagPi magazine and likes to explore science and data with Shake and Raspberry Pi.

magpi.cc

You'll Need

- Assembled Raspberry Shake (or you can access somebody else's data)
- Raspberry Pi
- Raspbian OS

In the previous tutorial, we looked at setting up Raspberry Shake, a geophone-based earthquake detector and checking out data with the web-based ShakeNet service: magpi.cc/shakenet.

Our shake has been running for a while now and we've gathered together some data on earthquake activity in our local area. We don't live in an earthquake-prone part of the world, but it's good citizen science and we can tap into other Raspberry Shake devices around the globe.

In this tutorial, we're going to take a closer look at the data provided by Raspberry Shake devices. We'll delve into how data is measured, stored, and what you can do with it. We're going to look at a helicorder and using Swarm to analyse live data.

Figure 1

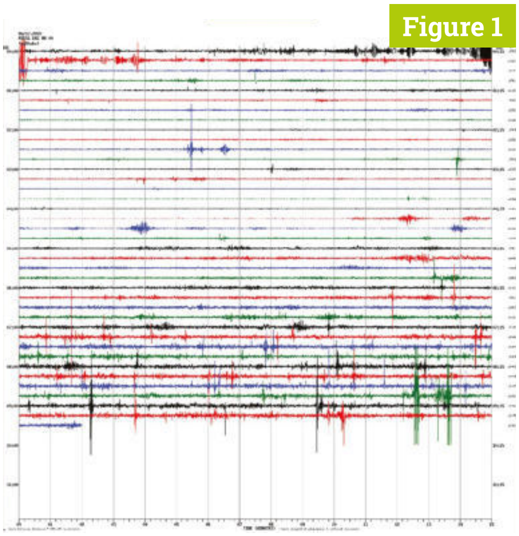


Figure 1 The Helicorder displays seismic activity for twelve hours. Each hour is divided into four horizontal lines (coloured black, red, blue, and green). Each coloured line represents a quarter-hour of activity

01 Open Raspberry Shake

We're going to use two Raspberry Pi devices in this tutorial. The first is used in our assembled Raspberry Shake unit (currently sitting in our conservatory). The second is used to remotely access the Raspberry Shake and investigate its data. You can perform much of this tutorial using another Linux, Windows, or Apple macOS computer if your only Raspberry Pi is being used as Raspberry Shake.

With both your computer and Raspberry Shake on the same network, Start by opening the web browser and navigate to:

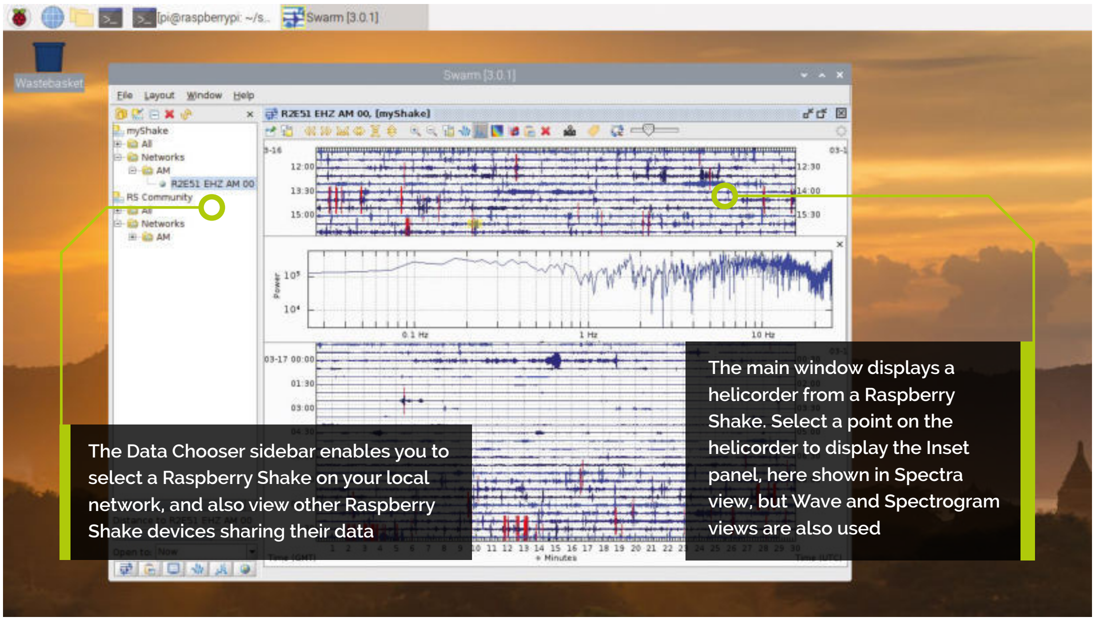
```
rs.local/
```

This will open the Raspberry Shake Config window.

02 Open helicorder

Click on the helicorder icon (shaped as four wavy lines) in the bottom left of the window. You will see 14 blue links, each with a date-stamp followed by either (12) or (00). These represent Raspberry Shake readings for the last seven days, split into 12-hour blocks. The ones marked '(00)' are for the morning hours (midnight to midday), while the '12' ones are for evenings (midday to midnight).

Click on one of the links to view the helicorder for that time frame. It'll look like the image in **Figure 1**. The latter shows seismic data for 12 hours. Each line represents 15 minutes of recording, and the lines vary in colour (black, red,



The Data Chooser sidebar enables you to select a Raspberry Shake on your local network, and also view other Raspberry Shake devices sharing their data

The main window displays a helicorder from a Raspberry Shake. Select a point on the helicorder to display the Inset panel, here shown in Spectra view, but Wave and Spectrogram views are also used

blue, and green). Down the left, you will see the time local to your area; to the right, you'll see UTC (Coordinated Universal Time). Our Shake unit is located near the Prime Meridian, so both times are the same.

Look along the lines to view seismic activity. Lines will typically be stable, and more motion could indicate somebody walking nearby, or other motion. Take a look this Maryland Geological Survey website for more information on how to read helicorder records: **magpi.cc/readhelicorder**.

03 Adjust helicorder scaling

We found our helicorder settings initially too intense (see **Figure 2**); conversely, you may find the helicorder on your Shake to be too mild. Either way, you won't be able to determine between different periods of intensity.

Head into your Shake settings and adjust the Helicorder Scaling Value to fine-tune your settings. Click on the Settings icon at **http://rs.local** and choose the Data tab. Adjust the Helicorder Scaling Value. The default setting is 0.5; adjust it down to 0.1 if the display is too intense, and up to 1.0 if it is too mild. You can fine-tune the levels to your taste as you go.

Click Save and Restart to put the new settings in place.

The helicorder will start displaying new recordings using the new scaling value, but will not retrospectively adjust the previous recordings. So you will need to wait until the end of your 12-hour recording for a fresh helicorder to display wholly adjusted results.

> "Each line represents 15 minutes of recording, and they vary in colour (black, red, blue, and green)"

04 Swarm

The helicorder is not updated in real-time. For real-time data, you'll need to use another app. There are many third-party apps available for data analysis, but Swarm (Seismic Wave Analysis / Real-time Monitoring) is the most commonly used by Raspberry Shake owners.

Swarm was developed by the USGS Volcano Hazards Program (**magpi.cc/usgs**) and is the most widely used seismological application in the world. Swarm is available for Linux, macOS, and Windows operating systems, and Raspberry Shake provides a version that is preconfigured for Shake devices.

In the case of Swarm, you can also connect to the Raspberry Shake Community server to see waveforms from all of the other Raspberry Shakes in the world.

Top Tip

StationCode

The StationCode is generated automatically by using the last four digits of the Raspberry Pi computer's unique hardware address (commonly referred to as the 'MAC' address) and appending them to the letter "R". **magpi.cc/stationcode**

Top Tip

Get the manual

Download the Swarm V3 User Manual and Reference Guide directly from the USGS website. **magpi.cc/swarmpdf**

05 Download Swarm

Visit **rs.local** in your web browser and click the SWARM Download button. Open a Terminal window and navigate to the **Downloads** folder:

```
cd Downloads/
```

Now unzip the downloaded **swarm** folder and move the unzipped folder to your home folder:

```
unzip swarm-3.0.1
mv swarm-3.0.1 ~
```

(If Swarm has been updated, replace the file name with the appropriate latest version.)

06 Open Swarm

Now open the **swarm** folder in your home folder and run the **swarm.sh** file script to start the program.

```
cd ~/swarm-3.0.1
sh swarm.sh
```

If you are using a Windows computer, you will need to install Java first (**magpi.cc/javawindows**) and then run Swarm by double-clicking the **swarm_console.bat** file.

07 Access your Raspberry Shake in Swarm

When you first open Swarm, it will display a blank blue window; to the left, a sidebar will display myShake and RS Community folders. Double-click myShake to reveal further subfolders, then click the '+' sign next to AM to reveal the StationCode for your Raspberry Shake. Ours is R2E51.

Double-click on the StationCode and the main window will display a helicorder. This time, however, the lines will be blue, with darker blue and red colours used to indicate heavy periods of activity. Each line represents a half-hour of activity, and the helicorder displays live data.

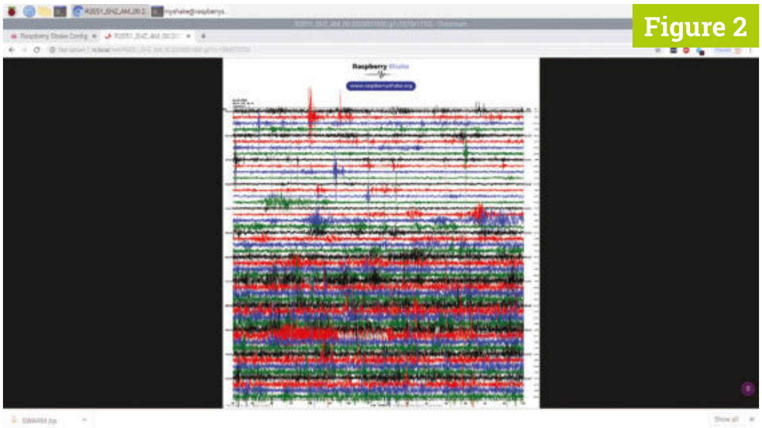


Figure 2

Figure 2 Our helicorder setting was too intense, requiring the Scaling Value to be reduced so values could be inspected

08 The inset window

Click on any part of the helicorder to open the Inset window. This shows a zoomed-in area of the helicorder. The first time you click on it, it will be in Wave view (this is the helicorder wave expanded to make it easier to view). Right-click with the mouse to switch to Spectra view; right-click again to view a Spectrogram. Icons in the Status bar above the helicorder are also used to change views.

"You can see the strength of seismic activity in specific GHz bands"

09 Spectrogram view

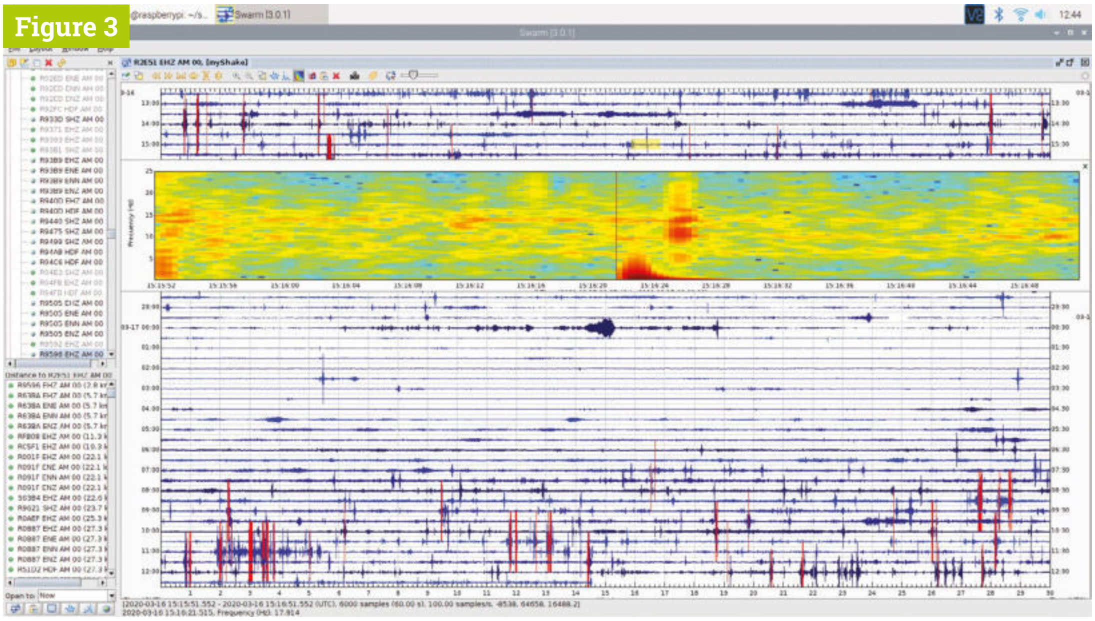
The Spectrogram view displays the frequency of waveforms concerning time and amplitude (or power). The X-axis (horizontal) of the spectrogram relates to time (as with a regular helicorder plot); the Y-axis (vertical) relates to the frequency of the wave. A third data point is displayed via the colour of the points on the graph: blue is for the weakest energy, and red is for the strongest. So you can see the strength of seismic activity in specific GHz bands. The Spectrogram is a very powerful tool for understanding the seismic activity.

Ben Ferraiuolo has written a great article called 'How to understand spectrograms', which can be used to get a better understanding of how to interpret this data (**magpi.cc/spectrograms**).

10 Helicorder view settings

All the settings for the helicorder view can be manipulated in the helicorder view settings dialog, which can be opened by clicking on the

Figure 3



[2020-03-16 15:15:51.552 - 2020-03-16 15:16:51.552 (UTC), 6000 samples (60.00 s), 100.00 samples/s, -8538, 64658, 16488.2]
2020-03-16 15:16:21.515, Frequency (Hz): 17.914

Figure 3 The Spectrogram displays the frequency of waveforms concerning time, and the colour of the data points indicates the amplitude (power) in that frequency

Helicorder View Settings button in the status bar. The 'X, minutes' option adjusts the length of time each horizontal waveform represents (the default is 15 minutes), while the 'Y, hours' option determines how many hours are represented on a screen (24 by default).

The Zoom option adjusts how many seconds are displayed in the Inset panel (the default is 30; set this higher to get a wider view of data when you click on the helicorder.)

The Clipping option determines the level at which the red threshold is exceeded. Finally, the 'Refresh, seconds' option is used to adjust the update frequency (by default, the heliograph is updated every 15 seconds).

11 Wave clipboard

You may find it useful to compare different sections of the helicorder, or the same time using different views: plot, spectrogram, and so on. The Wave Clipboard is used to hold different clips from several areas at once (even across different Raspberry Shake models).

Choose Window > Wave Clipboard to view the current selection (which should be empty).

Head back to the helicorder and click on an area of interest to open the Inset Panel. Right-click to get a View mode and click the 'Copy inset to clipboard' button. The Wave Clipboard will return to focus and now display the clipped wave. You can add multiple waves to the Wave Clipboard and remove them using the 'X' delete icon to the right of each wave.

12 View another Raspberry Shake

By now you should have a good understanding of how Swarm is used to distil seismic data. However, you can also use it to access other Raspberry Shake devices that are sharing their data publicly.

Double-click RS Community in the Data Chooser sidebar and expand (with '+') the Networks folder and the AM folder. This will display all Raspberry Shake devices on the network.

Choose your Raspberry Shake underneath myShake and the Data Choose will display a list of devices, sorted by distance to yours. Select one of the devices and click the Map icon (at the bottom of the Data Chooser) to view its location. Double-click the device (or click the helicorder icon at the bottom of Data Chooser) to view its data.

We are accessing a Raspberry Shake unit made with a Raspberry Pi 3B+ over a local network from a Raspberry Pi 4.

Part 03

Facial recognition with a magic mirror

Mirror, mirror on the wall, who is the fairest of them all? Decide with a smart face-recognising magic mirror

MAKER

PJ Evans

PJ is a writer, tinkerer, and enjoys installing Raspberry Pi Zero computers where no-one asked for them to be.

@mrpjevans

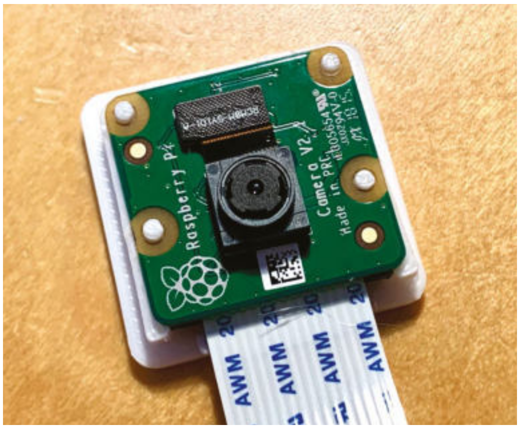
You'll Need

- A magic mirror! **magpi.cc/90**
- Raspberry Pi Camera Module **magpi.cc/camera**
- 3D-printed case (optional) **magpi.cc/cameracase**

In the past two issues, we've built and configured a magic mirror using the impressive MagicMirror² software. Now we have our mirror constructed and configured to our liking, it's time to go to the next level of personalisation. Wouldn't it be amazing if the mirror's display could be customised based on the person standing in front of it? Wouldn't it be even better if that could happen by the mirror recognising the person standing there? Turns out, that's well within the capabilities of a Raspberry Pi computer. We'll build multiple mirror configurations that appear when the right person is recognised.

01 Magic up a mirror

This project is an enhancement to the tutorials we covered in the previous two issues of *The MagPi*: 90 and 91. That means that you're going to need a magic mirror before we go any further.



The official Raspberry Pi Camera Module is a perfect choice for this project

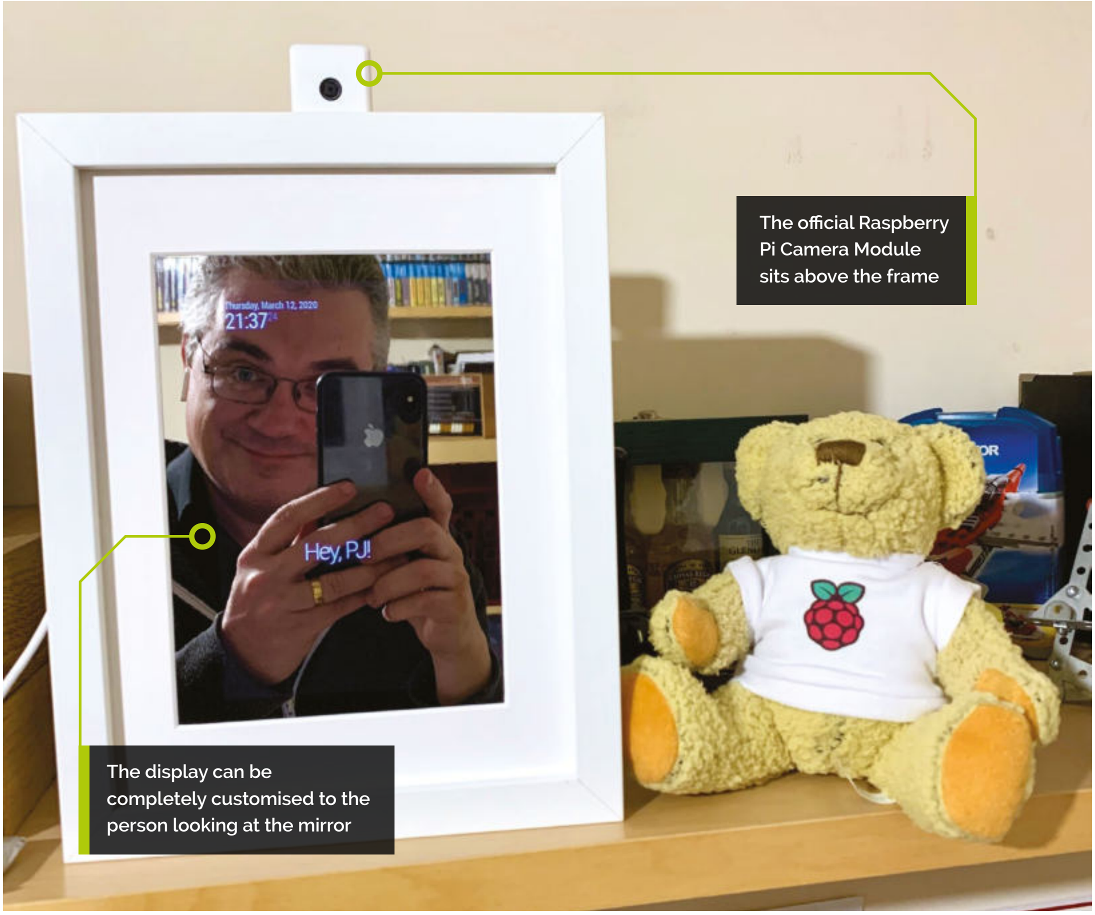
To recap, we made a 'budget' version using the Raspberry Pi official touchscreen, a small Ikea frame, and a generous amount of gaffer tape. Basic, but surprisingly functional. Having given it all a bit of a polish (software speaking) in the second article, we'll be building upon our current mirror. If you've gone your own way, that's great: the instructions will be roughly the same.

02 Casing the camera

If we're going to add the ability to customise the display based on who is standing in front of it, then we're going to need some vision. The easiest choice would be the official Raspberry Pi Camera Module. It's small, powerful, and easy to connect. How you want to mount the camera to your project is entirely your call. Here, we've 3D-printed a small case for the module so it fits in with its surroundings. The small case matches the frame and was easily secured with some strong tape. The more ambitious of you may consider mounting a camera directly in the frame.

03 Attach and enable the camera

Whatever camera solution you decide to use, you'll need to connect it to your Raspberry Pi computer. If you're using a Camera Module, attach the ribbon cable to the main board following the instructions here: **magpi.cc/picamera**. Once connected to the board, configure it by running `sudo raspi-config` from the command line and enabling the camera under 'Interfacing Options'. If you've gone with a USB webcam, see the instructions provided with it and check for Raspberry Pi compatibility before purchasing.



The official Raspberry Pi Camera Module sits above the frame

The display can be completely customised to the person looking at the mirror

Top Tip

Screen rotation

No luck with face recognition? Train, train, and train again. The more data, the more accurate the result.

04 Rotate the screen

The positioning of the camera may make you reconsider which way around you would like the mirror. If using the Camera Module, the provided ribbon cable requires that the mirror is portrait with the Raspberry Pi computer at the top. Changes in graphics systems with the move to Raspberry Pi 4 and Raspbian Buster mean the easiest way to set the screen orientation is within the Raspbian desktop itself. Stop the MagicMirror^2 software by entering this at the command line: `pm2 stop MagicMirror`. Then, in the Desktop environment, go to the menu, select ‘Preferences’, then ‘Screen Configuration’. Right-click ‘DSI-1’ and select ‘orientation’, then ‘right’. Now click the ‘tick’ to save.

05 Install OpenCV and more

Now we’re ready to get to work. To be able to recognise faces, we need some artificial intelligence. The amazing OpenCV software is not only free of charge but also superb at recognising objects, including individual human faces. Understandably, it’s a monster of an application, so we recommend a Raspberry Pi 4 with at least 2GB of memory. You can run it on older models, but it’s a lot more complicated. See the excellent **pyimagesearch.com** resource for more information. To install, run the following from the command line:

```
sudo apt -y update && sudo apt -y upgrade
sudo apt install libopencv-dev python3-opencv
```

06 Tool up

To be able to accurately identify someone’s face, we need to train our magic mirror. Using machine learning systems, we’ll take photos of the mirror’s users and create mathematical models of those faces so the mirror can recognise them in future. To do this, we’ll download some tools. Run the following commands:

```
cd ~
git clone https://github.com/mrpjevans/MMM-
Facial-Recognition-Tools.git
```

(If you get an error, run `sudo apt install git`, then try again.)

Top Tip

PyImageSearch

Want to know more about image processing and recognition using Python? There's no better resource than Adrian Rosebrock's **pyimagesearch.com**.

We now have a collection of tools to help train our mirror to the various faces it may encounter.

07 Training time

Now it's time to train our mirror. Run the following commands:

```
cd ~/MMM-Facial-Recognition-Tools
python3 capture.py
```

Choose whether to capture live photos from your webcam, or train from JPEG photos (you'll need to upload them in advance). You'll be asked for your name. Keep it simple. Your author chose 'pj', avoiding capitalisation, spaces, and special characters. You should now see your face on the mirror's display. Line up your face with how you think you'll be looking at the mirror ('in situ' is best) and press **ENTER** to take a snapshot of your face. Repeat this at least ten times before pressing **CTRL+C** to stop.

Finally, run `python3 train.py` to convert the images into mathematical data. When prompted for an algorithm, select LBPHF.

08 Testing time

To verify that the training worked, a testing script is provided. From the same directory, run these commands:

```
export FACE_ALGORITHM=1
export FACE_USERS=pj
python3 facerecognition.py
```

(Replace 'pj' with the name you chose.)

After the script has started up, have a look into the camera at the same angle you took the training photos. If all is well, you'll see reports that your face has been identified! This means the mathematical model has matched you with what it is currently seeing. If you don't get good results, try retraining or changing the lighting. Repeat the process until you're getting satisfactory results.

09 Install the module

Now we have our machine-learnt data, we can go ahead and install the actual facial recognition module. To do this, we need to install some code in the MagicMirror^2 **modules** directory:

```
cd ~/MagicMirror/modules
git clone https://github.com/mrpjevans/MMM-
Facial-Recognition
```

This installs all the files we need to add the module into the system. To complete the installation, we need to add dependencies:

```
cd MMM-Facial-Recognition
npm install
```

After a short time, all the files we need to run the module will be installed.

Finally, make a copy of the training data from Step 07 into the **modules** directory:

```
cp ~/MMM-Facial-Recognition-Tools/training.
xml ~/MagicMirror/modules/MMM-Facial-
Recognition/
```

10 Configure the module

Now we're all ready to add the module to our magic mirror. Just like last month's tutorial, we need to edit the **/config/config.js** file to add our module in. This is in JSON (JavaScript object notation) format, which is unforgiving of errors, so add things in carefully and if in doubt, make a copy of the file first so you can also 'roll back' should it be needed.

```
nano ~/MagicMirror/config/config.js
```

Scroll down the file until you see the 'modules' section. There will be a series of entries grouped by

{ and }, all of them grouped in [and]. You need to add the **list1.js** listing printed here into the [and] group. Each group of { and } needs to separated by a comma (except for the final one). Don't save the file just yet.

> After a few seconds, the system should restart and the 'default' modules are shown

11 Customise the modules

Have a look at the config section you've just added. Where it says 'users', make sure the names in the following square brackets match the ones you have trained for. Ours was 'pj' so it should read [‘pj’]. If you trained multiple people, separate them with commas: [‘alice’, ‘bob’].

Now, for each module you want to customise per person, add an extra parameter of 'classes'. This is a string with multiple entries separated by spaces. Add 'default' for always shown, 'everyone' for every recognised face, and specific names if that module is only to shown for that person. See the **list2.js** listing for an example.

12 Try it!

Now it's finally time to try out your mirror. Restart MagicMirror² by running this command:

```
pm2 restart MagicMirror
```

After a few seconds, the system should restart and the 'default' modules are shown. Have a look into the camera as you did for training and see if your modules appear. After a second or two, the mirror should refresh with your choice of display.

If you have any problems, use SSH to connect to your mirror and run `pm2 logs`. This will show you a live update of what's going on. Typically, any problems will be either not enough training or poor lighting. Just retrain by following the earlier steps and have another go. Soon you'll have a cool mirror that is unique to each of its visitors.

Thanks to Paul-Vincent Roll, author of the original MMM-Facial-Recognition module.

list1.js

Language: **JSON**

```

{
        module: 'MMM-Facial-Recognition',
        config: {
                // Array with usernames (copy and paste
from training script)
                users: ['pj']
        }
}
```

list2.js

Language: **JSON**

```
var config = {

        ...

        modules: [
                {
                        module: "clock",
                        position: "top_left",
                        classes: 'default'
                },
                {
                        module: "compliments",
                        position: "lower_third",
                        config: {
                                compliments: {
                                        anytime: [
                                                "Hey, PJ!"
                                        ]
                                }
                        },
                        classes: 'pj'

                },
                {
                        module: 'MMM-Facial-Recognition',
                        config: {
                                // Array with usernames (copy
and paste from training script)
                                users: ['pj']
                        }
                }
        ]

};
```

Part 01

Simon Safe Crack game

Use continuously rotating control for Raspberry Pi to create a rotary encoder game

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

Rotary encoders are the digital equivalent of a potentiometer, but unlike most potentiometers they can be continuously rotated. The big problem is that there is a lot of misinformation out on the internet about how to read them properly on a Raspberry Pi. So we thought we would take a look at how to really do this correctly. Along the way, we can also have a bit of fun with a fun safe-cracking simulator.

01 What is a rotary encoder?

There are two types of rotary encoder: absolute and incremental. The absolute type normally have lots of outputs, and are very expensive. An incremental type has two outputs: it is up to your computer to detect the direction of rotation, and count the pulses it produces in order to find out what rotation has occurred. Each type can be made with optical interrupters or switch contacts. The switch contacts are a lot cheaper, but come with problems of false contact events due to contact bounce. These can come with or without detents, which is a posh way of saying clicks.

02 How do they work?

See **Figure 1** – this shows a disc of metal with castellations on the perimeter. There are two spring contacts on the outer castellations, such that as the disc rotates, one and then both contacts will be touching the metal plate. There is a third, common contact, which is permanently attached to the metal plate. The contact that breaks connection with the plate first depends on the direction of rotation. There are four possible ways the two contacts can be connected or not connected, and these are shown in **Figure 1** as well. The sequence of connections is determined by the direction of rotation.

03 The KY-040 board

The low-cost KY-040 board consists of a rotary encoder with a built-in push switch. There are also on-board pull-up resistors for each contact. The schematic of this board is shown in **Figure 2**, so if you want to use a bare switch, this is how you wire it to be compatible with this tutorial. With this configuration, when you wire it

You'll Need

- KY-040 rotary switch **magpi.cc/rotaryswitch**
- 3 mm MDF or plywood for the box **magpi.cc/rotarywood**
- A 40 mm knob **magpi.cc/rotaryknob**

Figure 1

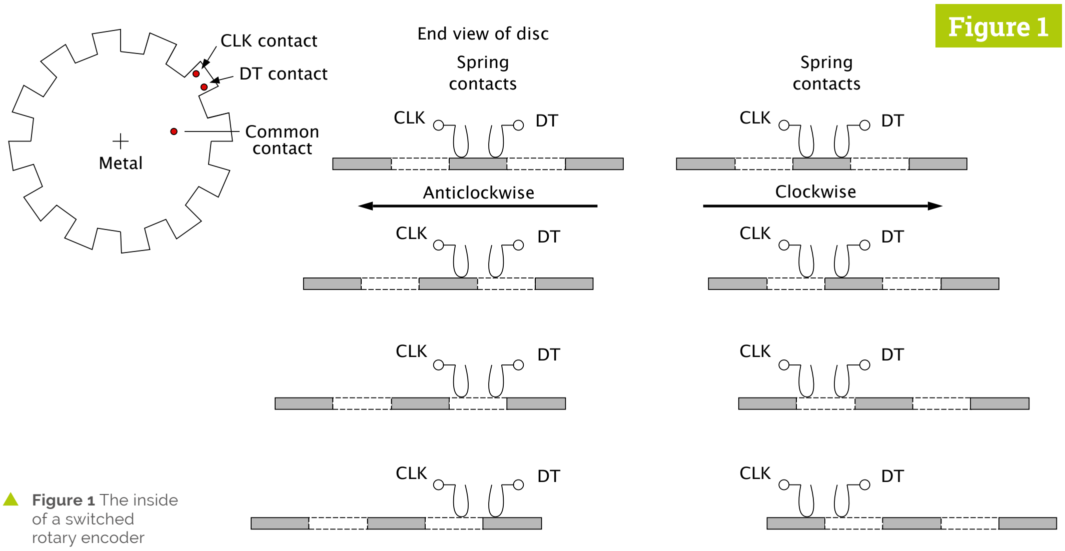
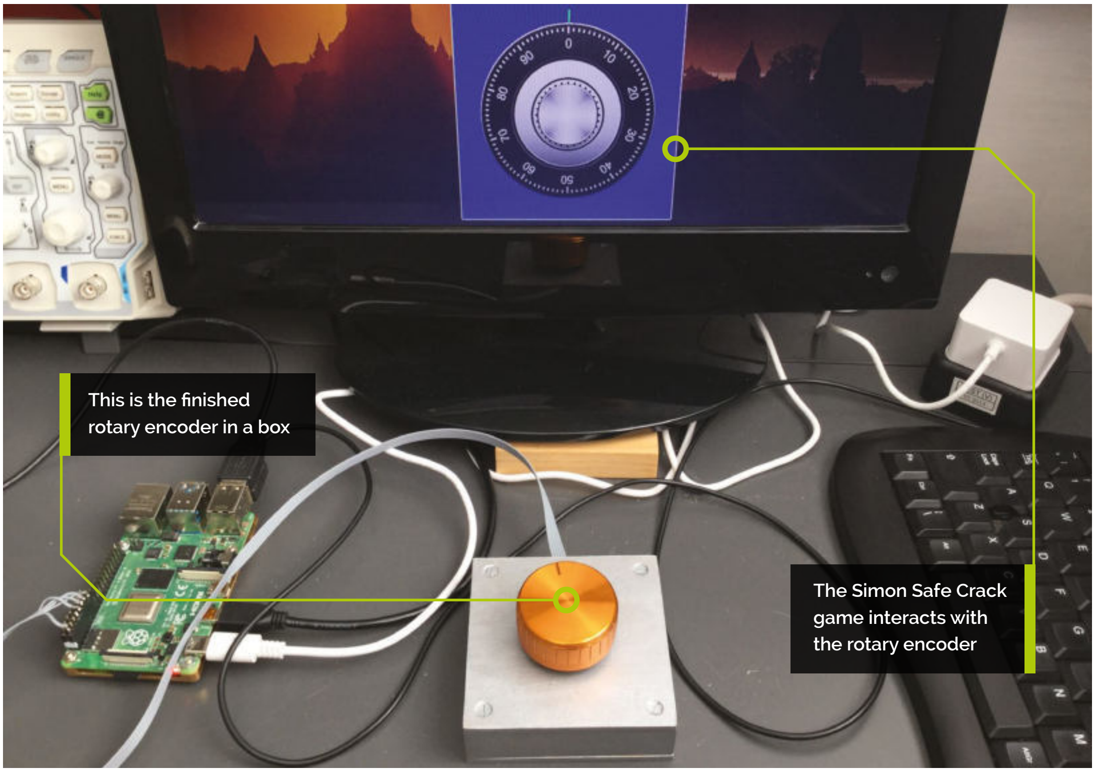


Figure 1 The inside of a switched rotary encoder



This is the finished rotary encoder in a box

The Simon Safe Crack game interacts with the rotary encoder

Top Tip

The pigpio version

While pigpio is pre-installed in Raspbian, this version will not work with Raspberry Pi 4. For that, you need version 74. To find out what version you have, enter `pigpiod -v -V` in a Terminal. See **magpi.cc/pigpio** for how to upgrade to the latest version.

up to a power supply, the sequence of signals you get is shown in **Figure 3**. Note that there are two types of encoders with detents, shown in **Figure 4** (overleaf). The detents are the resting place after a click; this board uses the first type, with the rest position giving the same output code.

04 Making the box

We made a simple 80×80 mm square box from 3 mm-thick MDF; the sides were 17 mm high and separated by 14 mm hexagonal threaded spacers, 8 mm from each corner. A 7 mm hole was drilled in the middle, and the encoder pushed through the hole. Where the anti-rotation pip met the board, we marked with pencil and drilled two 1 mm holes to form the slot. Then a 6 mm wide by 1 mm deep slot was made with a file on one of the sides. The sides were sanded off at an angle of 45° and glued together to form a square. The spacer holes were countersunk and the top was glued to the sides – see **Figure 5** (overleaf).

05 Finishing off

The box was painted silver, as a contrast to the gold knob, with PlastiKote B30 silver paint. A length of six-way ribbon cable was soldered to the KY-040 board and pushed through the slot. Finally,

Figure 2

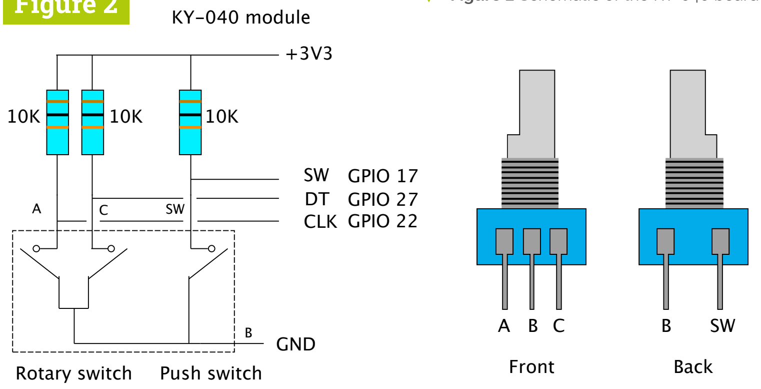
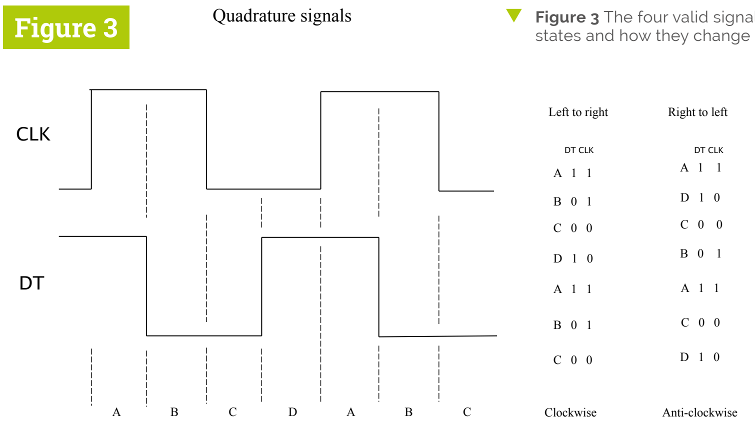


Figure 2 Schematic of the KY-040 board

Figure 3



Left to right

	DT	CLK
A	1	1
B	0	1
C	0	0
D	1	0
A	1	1
B	0	1
C	0	0

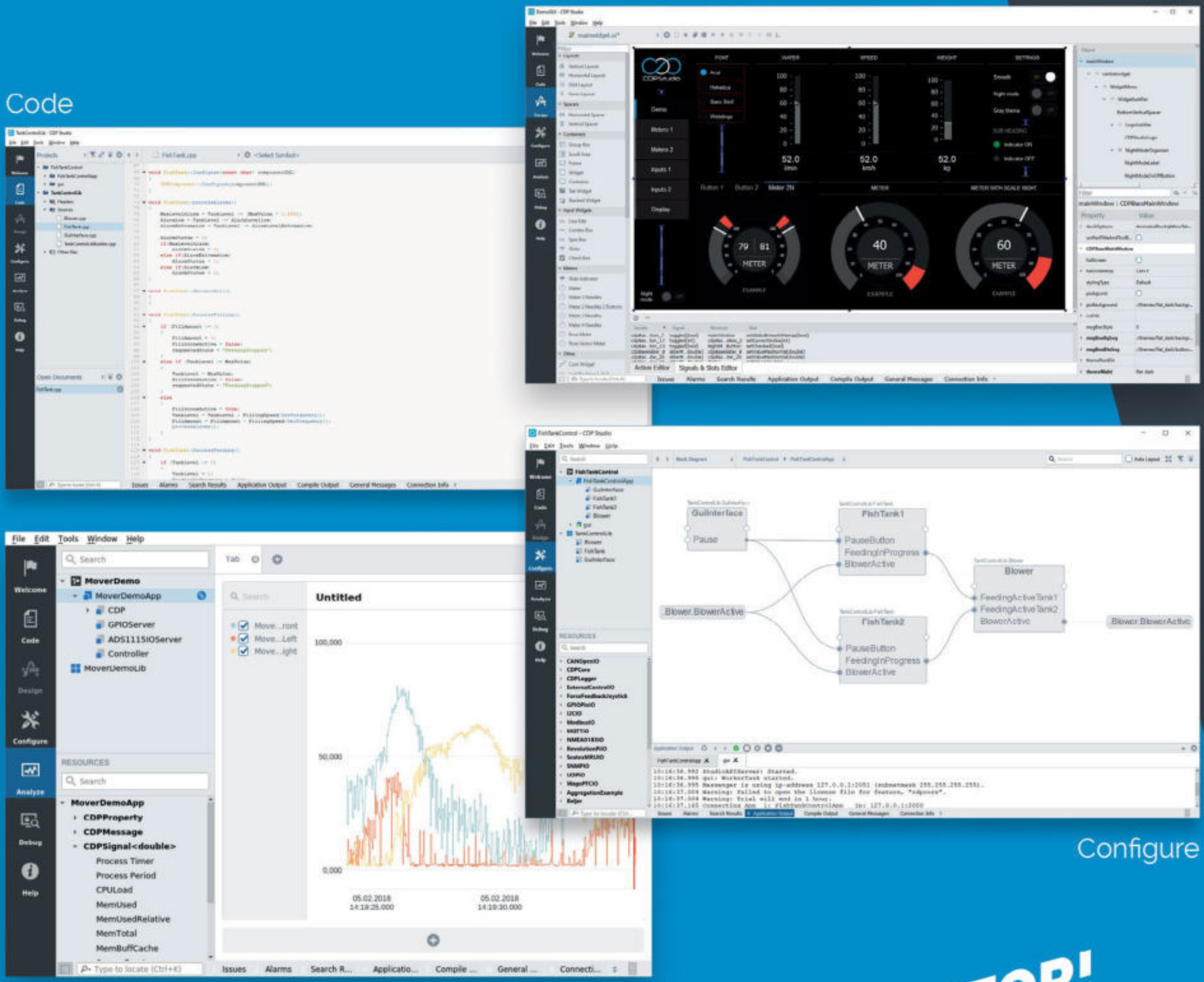
Clockwise

Right to left

	DT	CLK
A	1	1
D	1	0
C	0	0
B	0	1
A	1	1
C	0	0
D	1	0

Anti-clockwise

Figure 3 The four valid signal states and how they change



NEW GRAPHICAL EDITOR!

CDP STUDIO

Professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

CDP Technologies AS
Nedre Strandgate 29, P.O. Box 144
NO-6001 Ålesund, Norway
Tel: +47 990 80 900 • info@cdptech.com
www.cdpstudio.com

CDPStudio

Figure 4

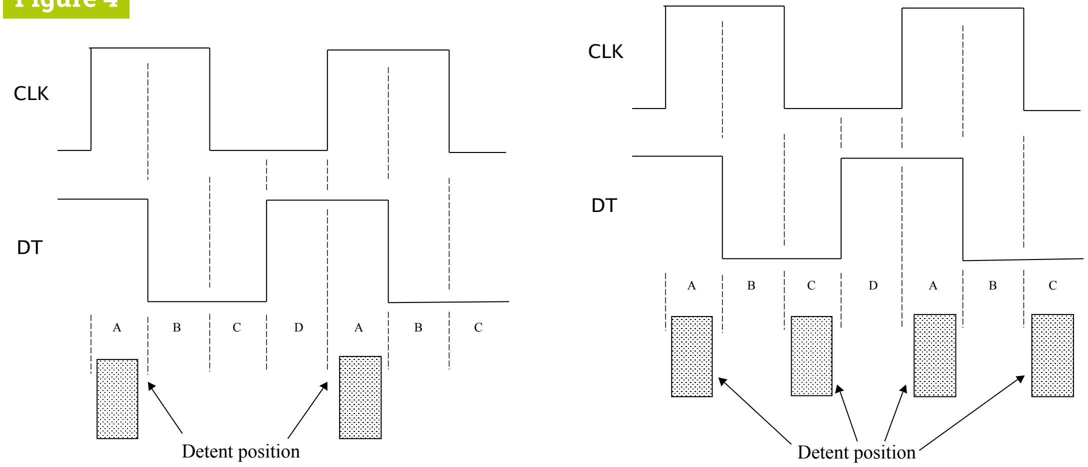


Figure 4 The detent resting position on two types of encoder

the other end was wired up to a 20-way header socket and wired up as shown in **Figure 6** (overleaf). You can wire up the board to any three GPIO lines, but you will have to change the pin assignment in the software to match. If your encoder has no thread, or you have no matching nut, you can mount the board to the top of the box with two 10 mm male-female spacers, as shown in **Figure 7**.

06 Contact bounce

While in theory the signals from the rotary encoder should look like the waveforms drawn so far, in practice it is not always the case. Consider the oscilloscope waveform shown in **Figure 8**. Here we see the falling edge of both signals is clean, as is the rising edge of the bottom trace. The rising edge of the top trace, however, shows considerable contact bounce, with many false edges before the switch settles down. Fortunately, there is a simple way to get round this, because in the normal course of operations you will always get signals first from one line and then the other. If you get two or more edges from the same line consecutively, then simply ignore those edges.

Figure 5

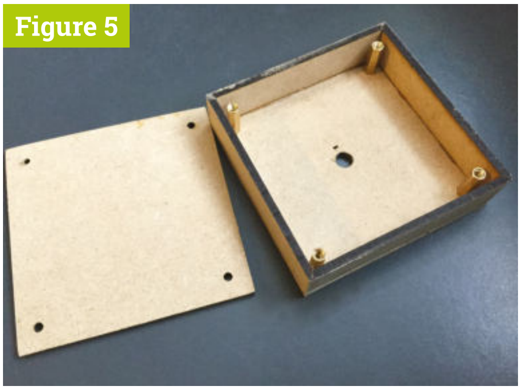


Figure 5 The MDF box construction

07 Software strategy

We need to have these edges trigger an interrupt, but the normal Python mechanism is way too slow to cope. Therefore, we have to use something faster. To the rescue comes the pigpio library. This is written in C, and can detect a transition within 1 µs (a microsecond) of it occurring, although the default 5 µs is fast enough. It can still take up to 10 ms for that event to percolate up through the system so that Python knows about it; however, pigpio buffers these events, so that although they might not arrive in real time, they do eventually arrive. Therefore, we do not miss any counts.

08 Class driver

We have written a class driver that will cause call back functions to be triggered from both the edges of the rotary signals, as well as the push switch. The push switch is optional, and we do not have to specify anything for this if we don't want them: simply miss out the pin assignments and callback functions when initialising an instance of the class. The code for the class driver is shown in **ky040.py** (download it with the other code files from **magpi.cc/pibakery**) and should be in the same directory as the script that's calling it.

09 Using the class driver

The **input_test.py** script is a small test program for using the class driver. This sets up an instance of the class that matches the wiring pin assignments we wired earlier. Once going, the callback functions will report any rotary and switch push action. Note: the switch pushes also return the time of the switch push or release as well as the

Top Tip

Wobbly encoders

Some of the low-cost suppliers don't always have the best-quality switches; we got some poor ones. In particular, if you make a click and get no response, this is because the detent has not returned the contacts to the correct place. A slight sideways motion will often free the contacts, and the click will then be registered.

DOWNLOAD THE FULL CODE:

magpi.cc/pibakery

Figure 6

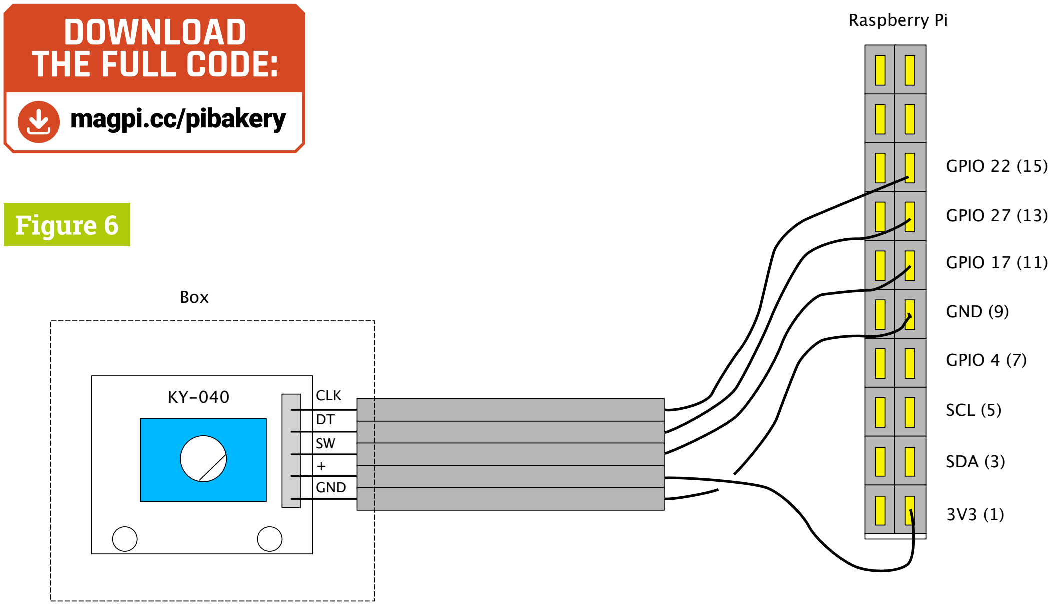


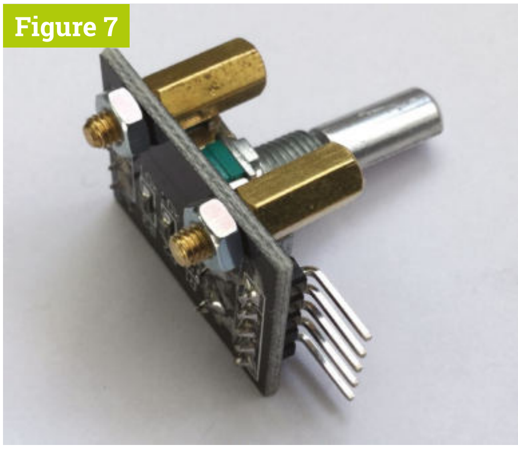
Figure 6 Wiring the KY-040 board to Raspberry Pi's GPIO connector

pin. If you run this code directly after bootup, it can fail to start pigpio correctly, and give you an error message. If this happens, just run the program again and it'll be fine for the rest of your session.

10 Simon Safe Crack

The **simon_safe_crack.py** script is the game itself. Here you have a combination lock dial, and the computer announces the combination using 'espeak'. If this is not installed, get it by entering `sudo apt install espeak` in a Terminal window. You then have to match the combination by turning the encoder the number of clicks and direction you were told to. For example: four left, ten right, one left. Note that the combination always ends with one. If you are successful at matching these instructions, an extra turn is added. This continues until you fail three times, and that signals the end of a round.

Figure 7



11 Customising the game

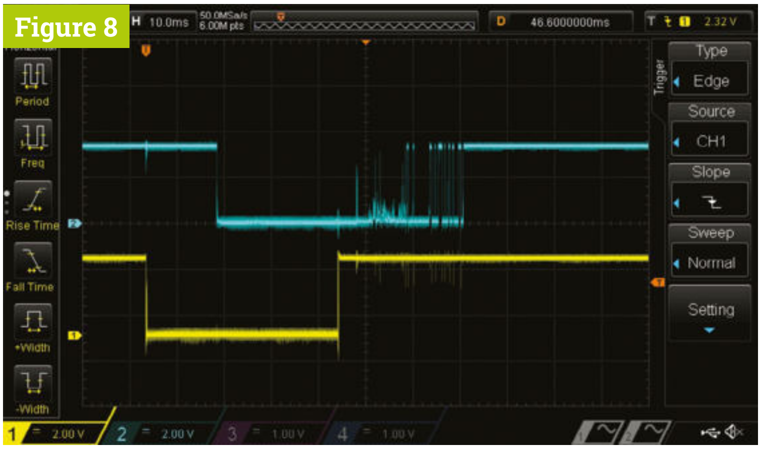
As the game is written under the Pygame framework, closing the window will finish the game, as will pushing the switch in the rotary encoder. In the `init` function, you can change the number of times you have to fail to finish a round with the `maxFails` variable. The `maxLength` variable defines how long the sequence has to get before you are considered to be cheating. In the `say` function, you can change the voice: the commented-out one is for a female voice. Finally, you can change the maximum number of clicks you are asked to move each time by the last number in the random number range in the `main` function.

Next month we will look at some more rotary encoder fun, using two encoders at the same time.

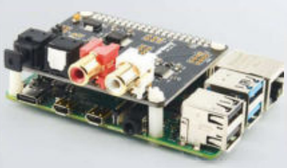
Figure 7 Mounting the board if you have no thread or nut

Figure 8 Possible contact bounce on the signals

Figure 8



Your Logo Here



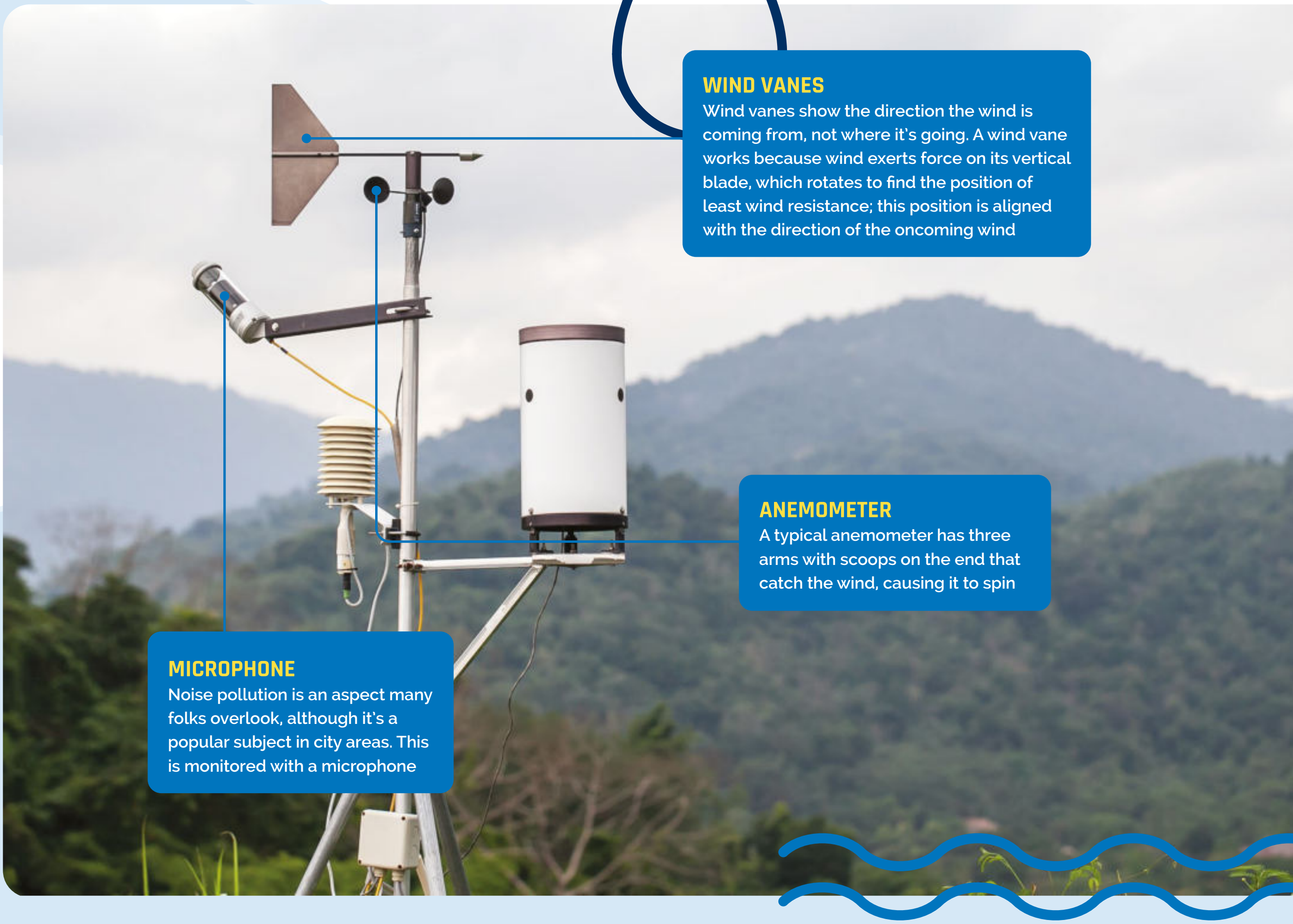
Track weather & pollution

WITH RASPBERRY PI

Air quality, humidity, and UV monitoring, as well as daily weather forecasting, all make for rewarding – and potentially creative – Raspberry Pi projects. **Rosie Hattersley** finds out more

Open data is a marvel. It allows us to learn and share so much about the world around us. If you're a weather obsessive or take a healthy interest in the state of the environment, being able to monitor changes – or simply predict what weird weather to expect next – is a great way to use your Raspberry Pi and coding skills. Over the next few pages we'll look at options if you're keen to learn how to set up your own weather station. Essentially, start small and focus on one environmental aspect, using an existing data source, before ramping up the number of sensors and reading your Raspberry Pi logs. In the case of air quality and pollution, tracking such changes has never seemed so relevant.

Making use of weather-related findings can also take you in different directions, from graphs, logs and user-generated maps to real-time tracking ideas with trigger warnings. As well as presenting project ideas you can emulate, we look at some of the most interesting Raspberry Pi-enabled weather tracking ideas to have caught our eye.



WIND VANES

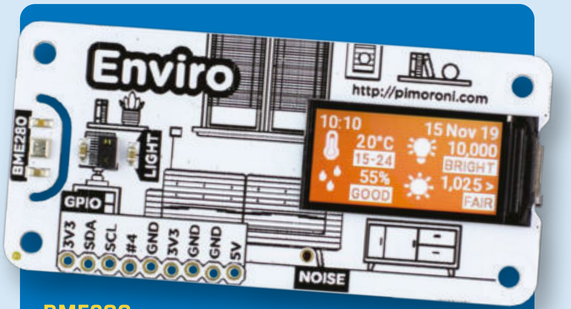
Wind vanes show the direction the wind is coming from, not where it's going. A wind vane works because wind exerts force on its vertical blade, which rotates to find the position of least wind resistance; this position is aligned with the direction of the oncoming wind

ANEMOMETER

A typical anemometer has three arms with scoops on the end that catch the wind, causing it to spin

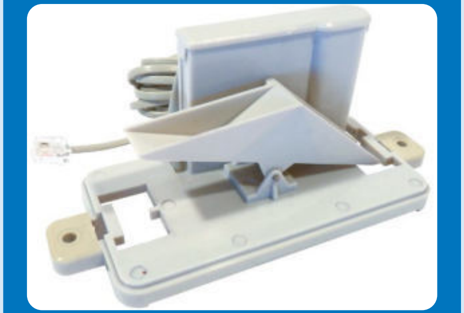
MICROPHONE

Noise pollution is an aspect many folks overlook, although it's a popular subject in city areas. This is monitored with a microphone



BME280

The Enviro board from Pimoroni features several sensors for weather and pollution: a BME280 for measuring temperature, pressure, humidity; LTR-559 light and proximity sensor; and a MEMs microphone. The Enviro + Air Quality version also includes a MICS6814 analogue gas sensor and a connector for an add-on particulate matter sensor



RAIN BUCKET

This rain gauge is basically a self-emptying tipping bucket. Rain is collected and channelled into the bucket

Weather & pollution tutorials

Learn to track the elements with Raspberry Pi, a few sensors, and these tutorials

Build your own weather station

The Raspberry Pi Foundation's Weather Station project is a classic, and pre-assembled kits were sent out to schools around the UK.

To make one at home, you'll need a Raspberry Pi with wireless LAN, plus some or all of these to create a sophisticated weather station: a BME280 combined pressure/humidity/temperature sensor, an anemometer, wind vane and rain gauge, a digital thermal probe, a breadboard, and some jumper wires.

You can pick and choose which types of weather and environmental changes you wish to track, and it is best to start with one or two sensors and move on up.

Data from the Raspberry Pi add-on sensors can be imported using straightforward Python code listed on each project page. For example, if you have a thermal probe, you can use it to compare the air temperature to that of the ground. Depending on the season, this may give very different readings.

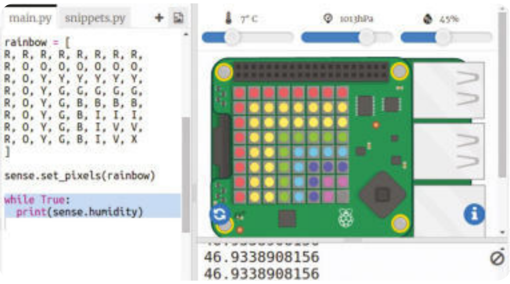
magpi.cc/weatherstation

Sense HAT rainbow predictor

There's something really special about seeing a rainbow. The Sense HAT's LED matrix helps improve your odds of seeing one by alerting you when conditions are right. That's when there's at least 80% humidity and the temperature is about 20°C. The instructions at **magpi.cc/rainbows** explain how to use a Trinket emulator to make your own virtual rainbow by assigning values to colours. Then you can use a Raspberry Pi Sense HAT (**magpi.cc/sense-hat**) to build the project for real.

You need to look up and select colour values using the RGB chart at **magpi.cc/trinketcolours**. When the Trinket's colours begin to make a rainbow, the weather conditions are also aligning to make a rainbow sighting more likely. A great way to add a reward for writing code.

magpi.cc/rainbows



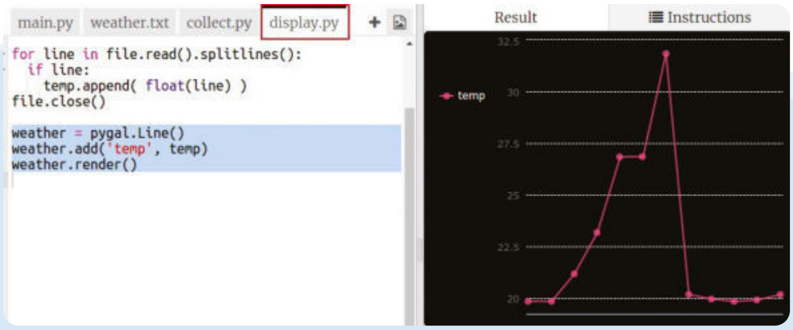
```
rainbow = [
R, R, R, R, R, R, R, R,
R, O, O, O, O, O, O, O,
R, O, Y, Y, Y, Y, Y, Y,
R, O, Y, G, G, G, G, G,
R, O, Y, G, B, B, B, B,
R, O, Y, G, B, I, I, I,
R, O, Y, G, B, I, V, V,
R, O, Y, G, B, I, V, X
]

sense.set_pixels(rainbow)

while True:
  print(sense.humidity)
```

```
46.9338908156
46.9338908156
```

▲ Plot temperature elements as colour values on a Sense HAT's LED matrix. When it displays a rainbow, a real one may well appear in the sky

Dress for the weather

Isn't it annoying when the sun's shining when you leave home but by the time you reach your destination you're facing a downpour? If you're heading off for a day out and need to check the weather at your destination, with OpenWeatherMap you can find out local weather conditions so you can dress appropriately. Sign up for a free account at **openweathermap.org/api**. Instructions given at **magpi.cc/dresswisely** show you how to use the Python module to import OpenWeatherMap data. Using the code listed at **magpi.cc/weathermap**, select your country or home town to get up-to-date weather information for your area. Once you know the forecast, you can write what-to-wear instructions that match the impending storm front or heatwave.
magpi.cc/dresswisely



```
for line in file.read().splitlines():
  if line:
    temp.append( float(line) )
file.close()

weather = pygal.Line()
weather.add('temp', temp)
weather.render()
```

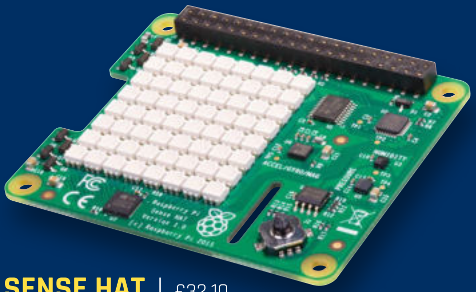
Weather logger

If you prefer to log weather details based on your own observations, the Weather Logger project (magpi.cc/weatherlog) can help. It makes use of information gleaned from your Sense HAT, and you plot the graph based on what its sensors report. First, you need to set the Sense HAT (or the online Trinket emulator) to check the temperature every five seconds. Once a set number of readings has been gathered, you use the Pygal Python module to create a graph showing its findings. Since you can set the parameters and specify temperatures, this project also works as a simulator for different weather conditions.
magpi.cc/weatherlog

Graphing the weather

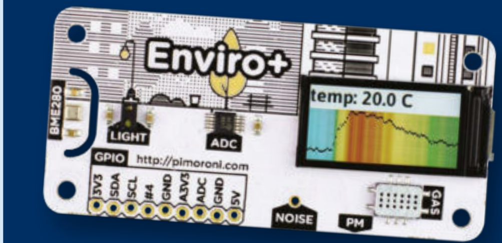
Get the impression rainfall is increasing and winds getting stronger? Using a weather tracker, you can record what happens each day and create a weather graph to back up your hunch. The 'Graphing the weather' project (**magpi.cc/weathergraph**) uses a RESTful API to pull in JSON data then uses the matplotlib Python library to create simple line graphs. Data is supplied from the Raspberry Pi Weather Database, information for which comes from the 1000 schools that were sent weather stations by the Raspberry Pi Foundation back in 2016. You can discover one that's local to you by using the 'Fetching the weather' resource (**magpi.cc/fetchweather**), then find out how to manipulate the data.
magpi.cc/weathergraph

ESSENTIAL KIT



SENSE HAT | £32.10

Featuring an accelerometer to measure speed, a gyroscope to sense direction, temperature, humidity and barometric pressure sensors, and a magnetometer, a Sense HAT can tell you exactly what's going on! Its 8×8 RGB LED matrix display provides a visual update.
magpi.cc/sense-hat



ENVIRO | £27.60 / £45

Available in two versions, the Enviro board is aimed at citizen scientists keen to monitor the world around them. Temperature, pressure, noise, and light levels are all logged, while the '+ Air Quality' model also tracks how clean the air is and encourages you to log polluting particulates in your neighbourhood.
magpi.cc/enviroplus

WEATHER STATION KIT | £145

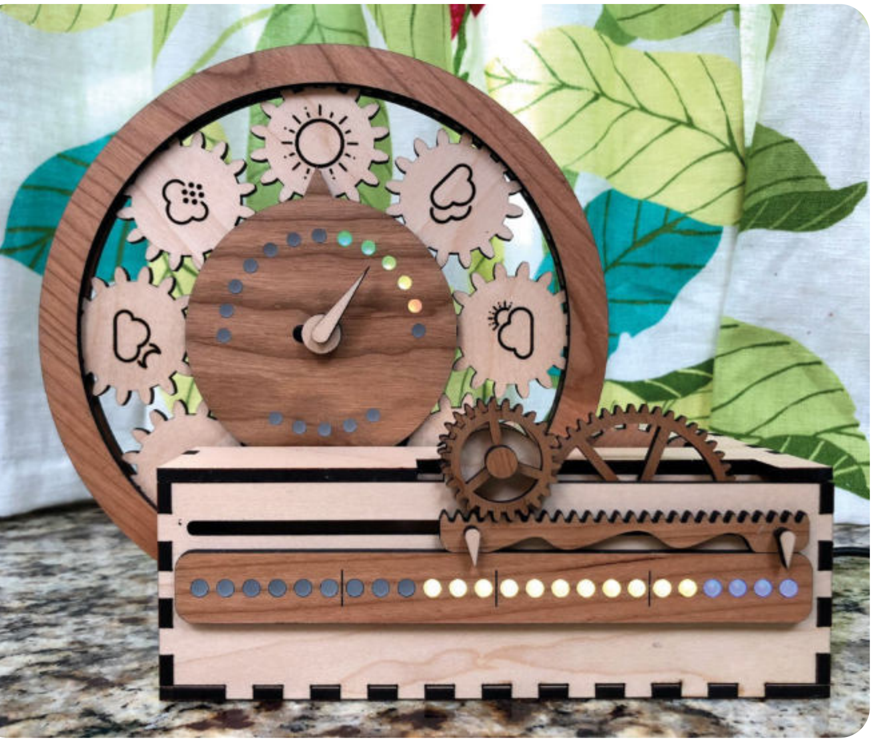
A classic wind vane weather station, The Pi Hut's kit logs wind speed/direction and rainfall, and is designed to be used in conjunction with electronic kits and single-board computers. Its meteorological focus is useful for logging environmental changes and for geologists, engineers, and urban planners.
magpi.cc/windvane

Learn to track weather & pollution

These inspirational projects make great use of easily accessible environmental information

Tide Clock Weather Project

Surely one of the most stylish presentations of weather effects, the Tide Clock Weather Project uses different coloured NeoPixel LEDs to display weather conditions and temperature based on details imported from the Dark Sky API. Weather symbols were specially laser cut and rotate using a gear mechanism. The clever clock also pulls in tidal information from the US NOAA and indicates whether the tide is high or low (and consequently, for its maker, when it's time to surf or kayak). A Speaker pHAT attached to Raspberry Pi Zero chimes at high and low tides.
magpi.cc/tideclock



Nemo-Pi

It's not just above ground that environmental sensors come in useful: coral reefs are being decimated by rising temperatures and human activity including damage from anchors. The Save Nemo Foundation in Thailand and Indonesia uses a Raspberry Pi-powered underground weather station, Nemo-Pi, to monitor temperature, pH, oxygen, and nitrogen levels. The solar-powered monitors attached to buoys report back to a central monitoring station. When the seas offer poor visibility, dive ships know it's not worth visiting, while pollution incidents can quickly be flagged and responded to, helping to preserve the coral reefs.
magpi.cc/nemopi

Smart Window Fan

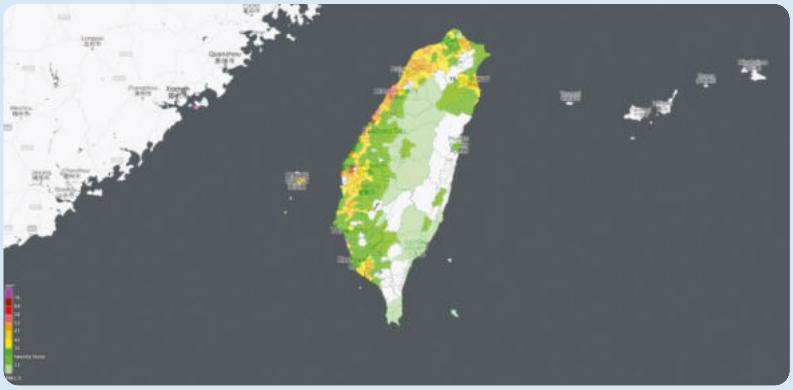
On a sticky summer's night, your repose is not exactly going to be restful if you have to keep getting up and switching the fan off and on again. Ishmael Vargas's alternative is genius: a smart window fan that uses local weather information to check whether the outside air is any cooler than inside and, only if it is, wafts external air into his muggy apartment. A Raspberry Pi Zero monitors conditions indoors, compares the temperature with that outside, and switches on a smart fan.

magpi.cc/windowfan

Cloudy-A

Much like the fantastic Weatherman project we featured in issue 90, the weather forecast lamp pulls weather in data using the Weather Underground API and presents it in a visual form. Cloudy-A is an internet-connected lamp that changes colour depending on the weather. The fact it also resembles a cloud says a lot about what the prevalent weather conditions are likely to be! The project has more than a nod to Blue Peter about it: the cloud's core is a five-litre rectangular plastic drinks bottle with two LED strips taped inside it, while the cloud is made from cotton wool.

magpi.cc/cloudlamp

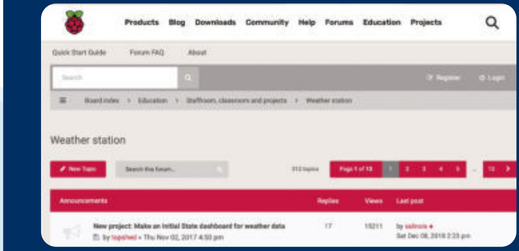


PM2.5 AirBox Weather & Pollution Project

If you've read about the Raspberry Shake seismographic sensor (magpi.cc/shakewlan), you'll appreciate how citizen science can be used to gather such data and create a useful map of earthquake tremors. Taiwan-based Raspberry Pi fans used a similar idea, making use of their country's PM2.5 Open Data Portal location-aware sensing system (aka LASS) to accrue information on localised air quality. New users are given instructions and help with setting up their PiM25 AirBox sensor, after which they can upload GPS and air quality data to the portal, from which the weather and pollution map is generated. See HackSpace issue 21 for how to make your own: **magpi.cc/airmonitor**.

magpi.cc/airbox

HELP & ADVICE



Products Blog Downloads Community Help Forums Education Projects

Quick Start Guide Forum FAQ About

Board index › Education › Staffroom, classroom and projects › Weather station

Weather station

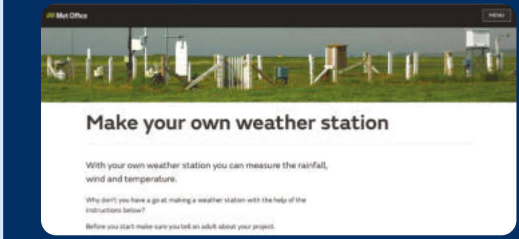
New Topic

Announcements	Replies	Views	Last post
New project: Make an Initial State dashboard for weather data by topshed » Thu Nov 02, 2017 4:50 pm	17	15211	by salinois Sat Dec 08, 2018 2:23 pm

WEATHER STATION FORUM

When building almost any type of weather tracking system, get setup and troubleshooting help from the friendly folks on the Raspberry Pi Weather Station Forum. Log issues here if you discover a site or service has a glitch. Details of successful projects and clever hacks are also encouraged.

magpi.cc/weatherforum



Met Office

MENU

Make your own weather station

With your own weather station you can measure the rainfall, wind and temperature.

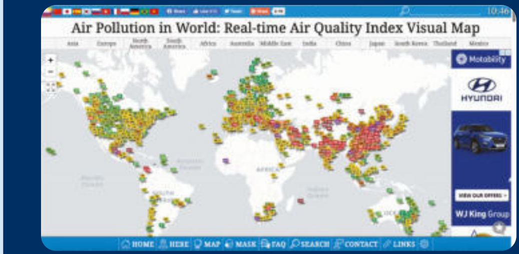
Why don't you have a go at making a weather station with the help of the instructions below?

Before you start make sure you tell an adult about your project.

MET OFFICE

Schools and clubs can give the Met Office a run for its money using its meteorological data to create your own weather station. Rainfall, wind, and temperature can all be tracked using the easy-to-follow instructions on its site. No Raspberry Pi or coding is needed for these quick-to-set-up projects.

magpi.cc/metoffice



Air Pollution in World: Real-time Air Quality Index Visual Map

TRACK AIR QUALITY WORLDWIDE

Find out how much UV, pollen, and other pollutants are currently affecting almost anywhere in the world right now. The colour-coded Air Quality Index can be viewed by pollution type and from country down to neighbourhood level. Click the Here tab to go straight to your nearest tracker. A shoo-in for a Raspberry Pi project?

aqicn.org/map/world

Build an air-quality monitor with Enviro

Monitor the environment by setting up an Enviro + Air Quality board, mounting it outdoors, and accessing the logged data on the web

MAKER

Phil King

Phil is Raspberry Pi Press's books editor, as well as being a sub-editor and contributor for *The MagPi*.

magpi.cc

Top Tip

Sticky strips

We used double-sided adhesive Command Strips (medium size) to attach the PM sensor and Raspberry Pi to the inside of the enclosure, but if you can get the hook-and-loop ones, they make it easier to remove the electronics later if you need to adjust anything.

Due to growing concerns about poor air quality and its effects on our health, many people are keen to monitor their own environment to check for spikes in often invisible pollutants. While you could create a DIY setup by connecting various discrete sensors to Raspberry Pi, Pimoroni's Enviro + Air Quality board makes it a lot simpler: just mount it on the GPIO header, install the software, and start monitoring. To check for microscopic particles in the air, such as from exhaust fumes, you can also plug in an optional particulate matter sensor.

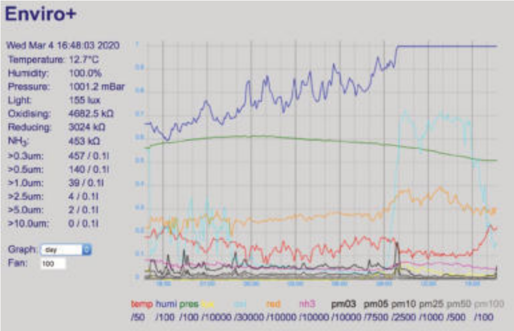
In this guide, we show you hot to set up your Raspberry Pi Zero and Enviro + Air Quality board, mount it in a weatherproof enclosure outdoors, and use Chris Palmer's EnviroPlusWeb software to log the data and access it a web dashboard remotely from another computer.

Note: This guide is based on Pimoroni's online tutorial for using Enviro with the Luftdaten citizen science project: **magpi.cc/luftdaten**.

01 Mount the Enviro

If you're using a Raspberry Pi Zero W, then you'll need to have a 40-pin male GPIO header soldered onto it. Alternatively, you can use a Raspberry Pi Zero WH, which already has a pre-soldered header.

Attach the Enviro board's female header to Raspberry Pi's GPIO pins. To make sure the board won't wobble around, you can use a couple of metal stand-offs the secure the edge furthest from the GPIO header to Raspberry Pi.



Enviro+

```
Wed Mar 4 16:48:03 2020
Temperature: 12.7°C
Humidity:    100.0%
Pressure:    1001.2 mBar
Light:       155 lux
Oxidising:   4682.5 kΩ
Reducing:    3024 kΩ
NH3:         453 kΩ
>0.3um:      457 / 0.1l
>0.5um:      140 / 0.1l
>1.0um:      39 / 0.1l
>2.5um:      4 / 0.1l
>5.0um:      2 / 0.1l
>10.0um:     0 / 0.1l
```

Graph: day
Fan: 100

▲ The web dashboard enables you to view the logged data over different time periods; this is the day view

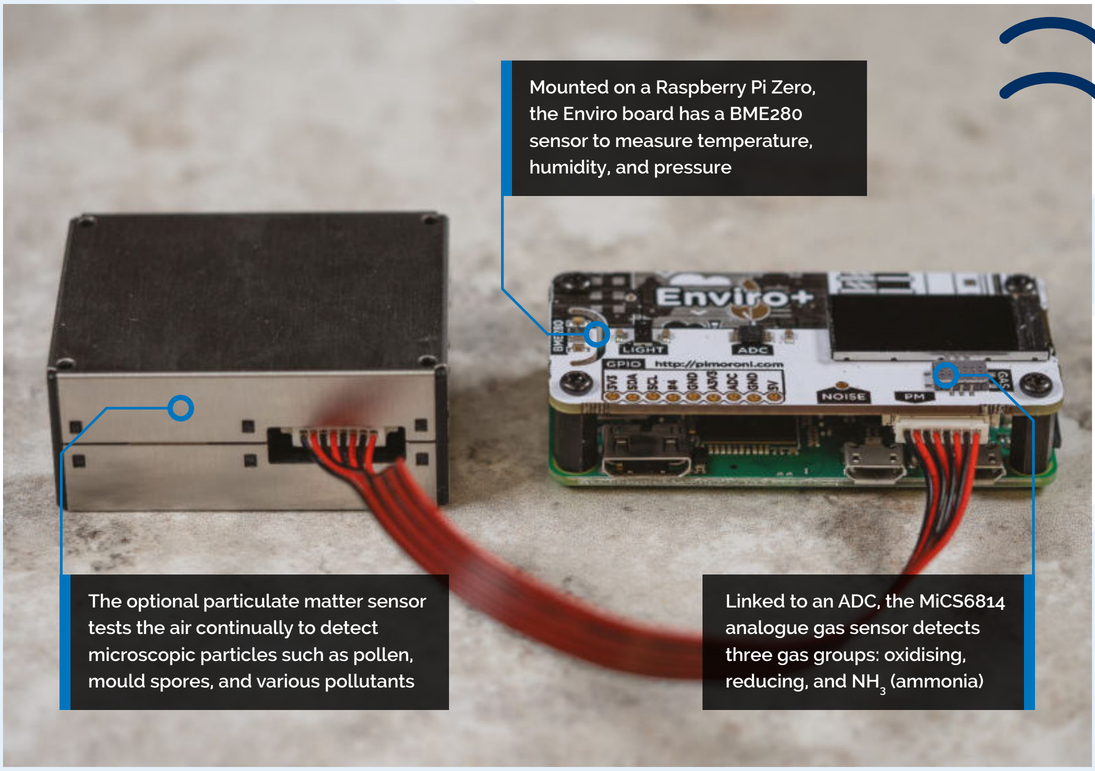
02 Attach the PM sensor

If you have the PMS5003 particulate matter sensor, connect the supplied cable to it – it'll only fit one way round, so don't force it. Plug the other end of the cable into the connector on the underside of Enviro board (below the PM label on the top side), again making sure that the cable is going in the right way round.

03 Boot up Raspberry Pi

To make sure that you have sufficient power, we advise using the long USB-A to micro-USB cable in the final build before putting your setup in the outdoor enclosure.

You'll need a microSD card with Raspbian Buster installed – while it's possible to use the Lite version, we'll assume you have the full version 'with desktop and recommended software', as it has the dependencies we need for the software.



Mounted on a Raspberry Pi Zero, the Enviro board has a BME280 sensor to measure temperature, humidity, and pressure

The optional particulate matter sensor tests the air continually to detect microscopic particles such as pollen, mould spores, and various pollutants

Linked to an ADC, the MiCS6814 analogue gas sensor detects three gas groups: oxidising, reducing, and NH_3 (ammonia)

Photos courtesy of Pimoroni

If it's not a fresh install, do a `sudo apt update` to make sure it's up to date.

Connect your Raspberry Pi to your wireless network, as usual, and enable SSH in the Interfaces tab of the Raspberry Configuration tool, so you can access it remotely later.

> EnviroPlusWeb logs data from the Enviro board's sensors and generates a Flask web server

04 Install Enviro library

To use the Enviro board, you'll need to install its Python library. Open a Terminal window and enter the following commands:

```
 git clone https://github.com/pimoroni/
enviroplus-python
 cd enviroplus-python
 sudo ./install.sh
 sudo pip install smbus2
```

Once that's all done, enter `sudo reboot` to restart your Raspberry Pi to apply the changes. The install script enables I^2C, SPI, and serial, disables the serial console, and enables the mini UART interface that Raspberry Pi uses to talk to the PMS5003 particulate sensor.

To check that everything is working correctly, run the all-in-one example:

```
cd examples
python all-in-one.py
```

Tap your finger on the board's light sensor to cycle through data from different sensors being displayed on its LCD. When you're happy it's all working, press **CTRL+C** to stop the program.

05 Install EnviroPlusWeb

You can now install the EnviroPlusWeb software, which logs data from the Enviro board's sensors and generates a Flask web server so you can view a dashboard remotely. From a Terminal window, enter:

```
 cd
 git clone https://github.com/nophead/
EnviroPlusWeb.git
```

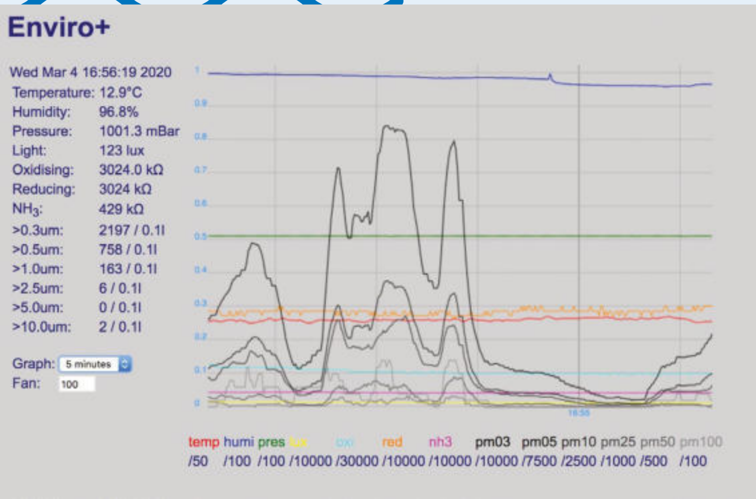
To test it out, change to its directory and run the **app.py** program:

Warning!
Outside mounting

When placing your air-quality monitor outdoors, take care if using a ladder. Also, make sure you have the property owner's permission.
magpi.cc/ladders

You'll Need

- Raspberry Pi Zero WH (or W with GPIO pins soldered) **magpi.cc/pizerow**
- Enviro + Air Quality **magpi.cc/enviroplus**
- PMS5003 particulate matter sensor **magpi.cc/pmsensor**
- Metal stand-offs (optional)
- Long USB-A to micro-USB cable (2m to 5m) **magpi.cc/microusb5m**
- 2 × Square 112.5° downpipe bends **magpi.cc/pipebend**
- 2 × Command Strips (medium) **magpi.cc/cstrips**
- Duct tape
- Cable ties (long)



Enviro+

Wed Mar 4 16:56:19 2020
Temperature: 12.9°C
Humidity: 96.8%
Pressure: 1001.3 mBar
Light: 123 lux
Oxidising: 3024.0 kΩ
Reducing: 3024 kΩ
NH3: 429 kΩ
>0.3um: 2197 / 0.1l
>0.5um: 758 / 0.1l
>1.0um: 163 / 0.1l
>2.5um: 6 / 0.1l
>5.0um: 0 / 0.1l
>10.0um: 2 / 0.1l

Graph: 5 minutes
Fan: 100

A Flask web server showing real-time and logged data can be viewed remotely from another computer

```
cd EnviroPlusWeb
sudo python3 app.py
```

Note that if you don't have a PM sensor connected, you'll get an error at this point. In this case, you'll need to use your preferred editor (e.g. Thonny, Nano) to remove – or comment out – the relevant lines (29, 49, 125 to 141, and 151 to 156) from **app.py**. You may also want to remove the relevant sections from the **readings.html** and **index.html** files in the templates directory.

If everything is working, you'll see the following in the Terminal when you run **app.py**:

```
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development
server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
```

Top Tip

Weatherproofing

Duct tape should prove sufficient to stop water getting into the enclosure. To make it extra watertight, however, you may want to seal the join with silicone sealant first, before applying the tape.

Figure 1

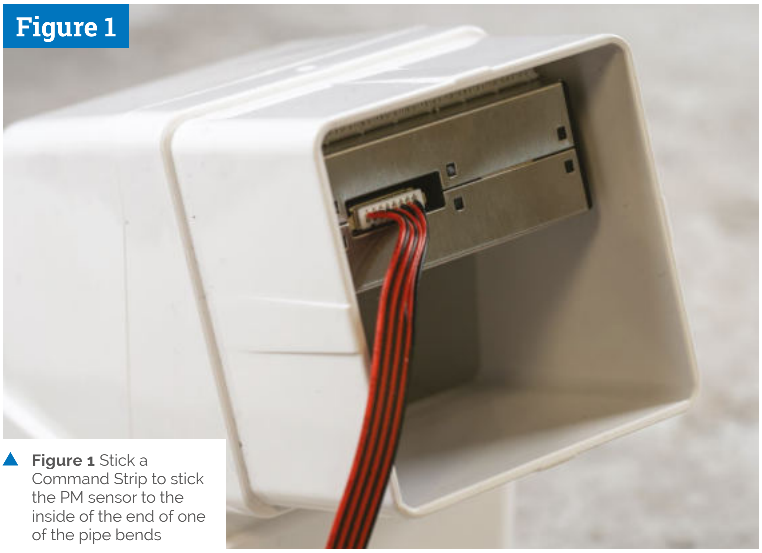


Figure 1 Stick a Command Strip to stick the PM sensor to the inside of the end of one of the pipe bends

06 View web dashboard

To view the dashboard created after running **app.py**, you simply need to visit your Raspberry's Pi's IP address (e.g. 192.168.1.76) in a web browser on another computer connected to the same network.

To discover the correct IP address, enter `hostname -I` in a Terminal window on your Raspberry Pi, or visit your wireless router's homepage (e.g. 192.168.1.254) and view the devices list.

Note: You may also want to reserve a static IP for your Raspberry Pi, so that the web dashboard address doesn't change if it's rebooted. This is most easily done in the settings of your router if it supports this. If not, edit the **/etc/dhcpcd.conf** file – see **magpi.cc/staticiphelp** for details.

> “To view the dashboard created, you simply need to visit your Raspberry's Pi's IP address”

07 Fine-tuning the graph

The web dashboard that's generated shows current data for the sensors on the left and a colour-coded graph of the readings. By default, this is for the past five minutes, but you can use the drop-down on the left to change it to day, week, month, or year.

If you find that some of the sensor readings are going off the top of the graph (or are always very low), you can change the min/max ranges for each one in the **index.html** file (lines 78 to 90) in the **templates** directory. For example, for the reducing gases reading, we changed it from a max of 1000 to 10000:

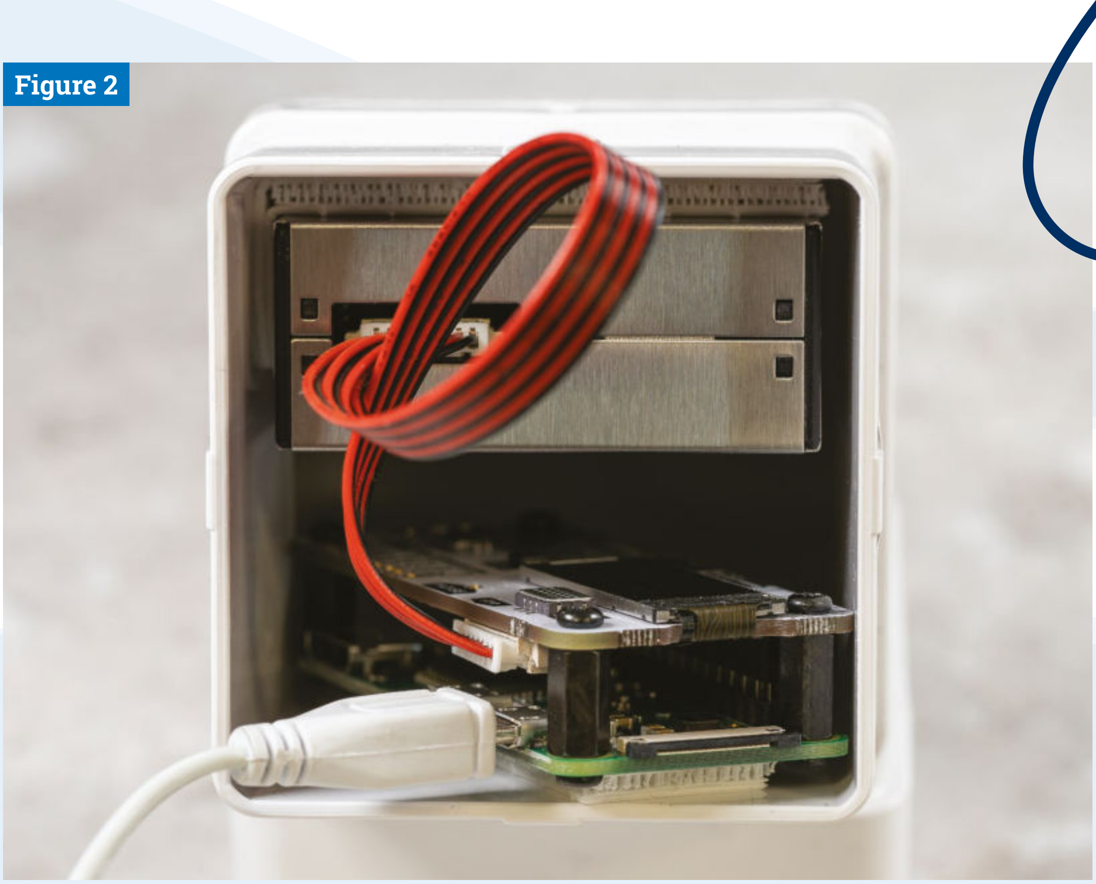
```
  {name: "red",   colour: "darkorange", min:
0, max: 10000}
```

08 Automatically run at bootup

Before you put your air-quality monitor in the outdoor enclosure, you'll want to get it to run the **app.py** script automatically whenever Raspberry Pi boots up. For this, we'll create a crontab job. In a Terminal, enter:

```
crontab -e
```

Figure 2



Photos courtesy of Pimoroni

◀ **Figure 2** Carefully position your Raspberry Pi Zero and Enviro at an angle so there's enough room for the power cable when the pipe bends are slotted together

At the bottom, enter a new line:

```
 @reboot sudo python3 /home/pi/EnviroPlusWeb/
app.py &
```

Press **CTRL+X** and then **Y** to save. Now, when you reboot Raspberry Pi, it should run automatically – check that it works and you can access the web dashboard.

Note: When the program is run automatically, the resulting data folder (containing JSON files of the logged data) is located in the **/home/pi** directory rather than the EnviroPlusWeb one. Chris says that a day's worth of data is about 62kB, so about 22MB per year, so after about 136 years it will fill the microSD card!

09 Build the enclosure

Now all the software is set up and tested, it's time to build the outdoor enclosure. First, shut down your Raspberry Pi and unplug it from the power supply.

Stick a Command Strip to the bottom of the PM sensor, then peel off the backing on the other side. Take the smaller, male end of one of your downpipe bends and stick the PM sensor inside the top of it (**Figure 1**).

Attach another Command Strip to the bottom of your Raspberry Pi Zero, then peel the backing from the other side. Now you need to stick your Raspberry Pi Zero at the bottom of the pipe, below the PM sensor, at an angle (**Figure 2**) so that you can attach the power cable with enough clearance to attach the female end of the other pipe bend.

Try fitting the two pipe bends together so that the female end slides all the way in, but don't force them if they won't go. Now boot up Raspberry Pi again to check that the script is running and everything is working. If so, stick some duct tape over the join between the two pipe bends to stop water getting in.

10 Mount the enclosure outdoors

The final step is to put your air-quality monitor outside. We used long cable ties (you can link two together for extra length if needed) to secure our enclosure to the underside of some conservatory guttering, but you may want to attach yours to a drainpipe. Either way, cross the cable ties diagonally to keep it firmly in place.

We ran our long USB power cable through a small window to reach our indoor power socket, but if you have a weatherproof outdoor socket, that's even better. M

Top Tip

Wireless signal

With the windows shut, we found that the strength of our air-quality monitor's wireless connection to our router (on the opposite side of the house) was a little weak and unreliable, with around 30/70 link quality – you can test this by SSHing in and entering the `iwconfig` command. Placing a plug-in wireless booster (a TP-LINK RE200) in our conservatory solved this.

PLAY & CODE GAMES!
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GAMING WITH RASPBERRY PI
164 PAGES OF VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE
RETRO GAMING WITH RASPBERRY PI
AMAZE BALLS!

Adafruit 1.3" Color TFT Bonnet

▶ Adafruit ▶ **magpi.cc/adafruit13tft** ▶ £17 / $18

A compact TFT display with two buttons and a joystick. Perfect for gaming? Not so fast! By **Lucy Hattersley**

SPECS

DISPLAY: IPS TFT, 240×240 colour pixels, controlled over SPI

CONTROLS: 5-way joystick and two push-buttons

CONNECTION: Qwiic/STEMMA QT connector for I²C sensors

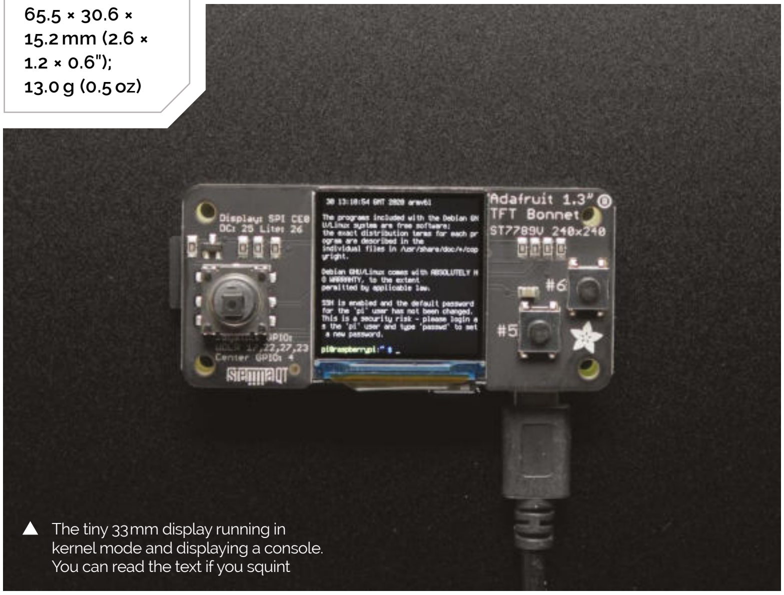
CHIP: ST7789 chipset

DIMENSIONS & WEIGHT: 65.5 × 30.6 × 15.2 mm (2.6 × 1.2 × 0.6"); 13.0 g (0.5 oz)

At first glance, the Adafruit 1.3" Color TFT Bonnet for Raspberry Pi reminded us of a classic tiny games console. A tiny 33 mm (1.3-inch) 240×240 display is framed by a five-way joystick on the left and two buttons on the right. The display is an IPS TFT type controlled over SPI.

Our immediate thought went to the old Game Boy Micro, and we considered putting it inside an old Nintendo game controller case and turning it into an ultra-micro games console.

This, it turns out, is a bad idea. As Limor Fried says in the launch video (**magpi.cc/bonnetvideo**): "I wouldn't use it for gaming, as it's very small and not that comfortable."



▲ The tiny 33 mm display running in kernel mode and displaying a console. You can read the text if you squint

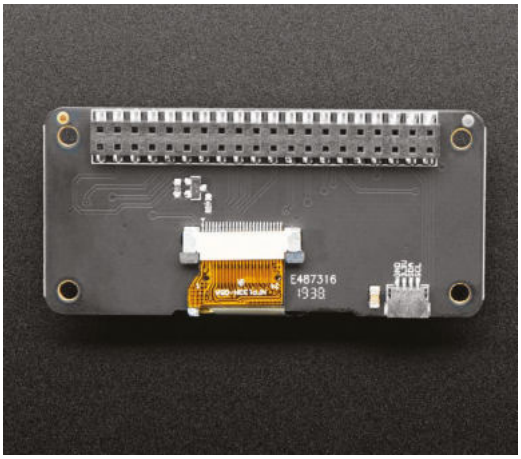
What's it for?

So, if a device so reminiscent of a classic console isn't for gaming, what is it for? The Bonnet shares a heritage with Adafruit's Mini PiTFT (**magpi.cc/minipitft13**). That device features the same square 33 mm display, but has just the two buttons. This Adafruit 1.3" Color TFT Bonnet comes with the five-way joystick to enable more complex interface interactions (the fifth direction is a push inwards, incidentally).

> “So, if a device so reminiscent of a classic console isn't for gaming, what is it for?”

Like the Mini PiTFT, the Adafruit 1.3" Color TFT Bonnet also features a Qwiic/STEMMA QT connector for I²C sensors. And this is where things become clearer. You can plug and play Adafruit's range of STEMMA QT devices (**magpi.cc/adafruitstemma**), which includes all manner of sensors: magnetometers, temperature, pressure, proximity, and so on.

It is in this area where the Adafruit 1.3" Color TFT Bonnet comes alive. What we have here isn't the



▲ Underneath the Bonnet is a STEMMA QT connector, enabling you to hook up a range of I2C sensors

heart of a portable games console, but an interface for a range of sensor projects.

In that spirit, it is something of a shame that the Bonnet covers the entire 40-pin GPIO header, unlike the Mini PiTFT which leaves 16 pins free. But the STEMMA QT connector provides your I/O needs.

Kernel or Python

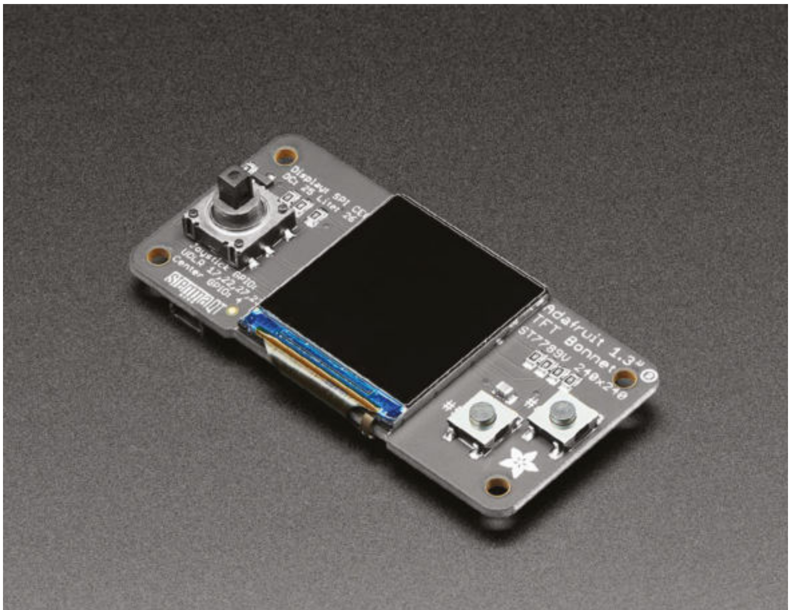
Setup of the Adafruit 1.3" Color TFT Bonnet was straightforward, although the linked instructions (**magpi.cc/tftguide**) are for the two-button Adafruit Mini PiTFT.

There are two approaches: use a script to install a kernel module, or use Python and the Pillow library (**magpi.cc/pillow**) to draw images on the display.

Crucially, you can't use both techniques at once. This is a shame as the kernel module is faster and, arguably, more fun. But once it's installed, you can't move on to using Python.

The kernel method is also more accessible method for beginners. Run a script and you can mirror the Raspbian with Desktop interface on the tiny 1.3-inch display.

It's fun but wholly impractical. Not quite ready to give up our dream of a teeny console, we attempted to install PICO-8 in this mode and while it did run, the experience was (as Limor claimed) too small.



▲ The screen is framed by a five-way joystick and two push-buttons, enabling you to create complex interfaces for your projects

The kernel installation script also enables you to run Raspberry Pi in a console mode. Text mode is better, if you have very good eyesight, but it's still lacking an effective use case. According to Adafruit, the Bonnet runs at 15 fps in kernel mode, so it is better suited to simple animations and video (neither of which is fun to watch on such a small display).

This leaves the second, more practical, option. Which is to follow the Python setup guide. This approach is (according to Adafruit) more stable and allows you to write Python code to control the display.

You need to install the Adafruit_Blinka library that provides CircuitPython support in Python. Once up and running, you can follow the tutorials to create your own display interfaces, and there are examples on the site. From here, you will be able to create interfaces that display information, and interact with your range of I^{2}C sensors.

The screen is sharp, and lovely to look at, but it's so small that it strikes us as a bit fiddly for anything complicated.

Verdict

We found this a bit of a head-scratcher, and we'd be interested to hear from makers who find it fits their use case. Unless you know exactly what you're going to make with it, we suggest going for the simpler two-button Adafruit Mini PiTFT.

7/10

Maker **pHAT**

▶ Cytron ▶ **magpi.cc/makerphat** ▶ £8 / $10

A board to help beginners get started with physical computing. By **Phil King**

SPECS

ON-BOARD COMPONENTS: Active buzzer, 3 × push-buttons, 8 × blue LEDs

BREAKOUT: Fully labelled 24-pin header, plus stacking header

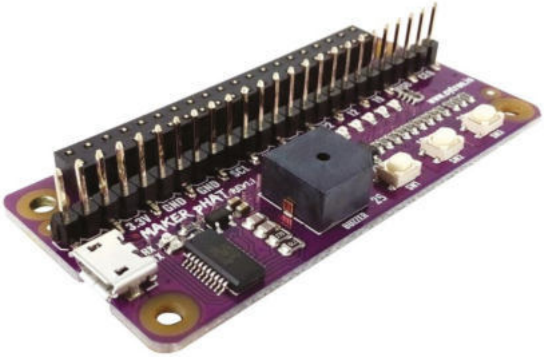
REMOTE CONNECTION: USB to serial module

When trying to learn how to use and program electronics with Raspberry Pi, learners – whether in a classroom setting or at home – face some typical problems. First, you need access to a keyboard and monitor. Second, there's the sometimes tricky business of attempting to create electronic circuits from separate components without getting it wrong and the possibility of 'magic smoke'. Then, if your project doesn't work, it's hard to tell whether it's because your wiring or code isn't right, or maybe both. This isn't helped by the difficulty of identifying the correct GPIO pins on Raspberry Pi, as they're unlabelled.

Cytron's Maker pHAT attempts to solve these issues and make it a lot simpler to get started with physical computing on Raspberry Pi.

Purple PCB

The cool-looking purple PCB has some common components already on board and connected to certain GPIO pins. Along with three small push-buttons, there's an active buzzer and eight tiny LEDs – we were slightly disappointed that they're all blue and not a range of colours. A nice touch is the inclusion of a fully labelled, 24-pin breakout header for connecting external components when you're done playing with the on-board ones.


▲ The board is the perfect size for mounting on a Raspberry Pi Zero, although stand-offs aren't included

While you can simply mount the board on your Raspberry Pi's GPIO header – with or without the supplied 40-pin stacking header – and start coding, the pièce de résistance is the inclusion of a USB to serial module. This enables you to connect the board to a laptop and control (and power) it and Raspberry Pi remotely from there, eliminating the need for a separate monitor and keyboard.

A comprehensive online manual (**magpi.cc/makerphatinfo**) explains how to install a special driver and get the serial connection working using PuTTY on Windows, though not on a Mac. For the latter, use Terminal and enter `ls /dev/cu.usbserial-*` to find the device number, then `screen /dev/cu.usbserial-XXXXXXXX 115200 -L` to log in (after pressing **ENTER** repeatedly). The manual includes a Python demo program, which makes use of GPIO Zero, to get you started – it even enables you to safely shut down Raspberry Pi by pressing two of the buttons together.

▲ The supplied stacking header provides full GPIO breakout and can be used to elevate the board on a full-size Raspberry Pi

Verdict

An inexpensive and well-designed board for physical computing newbies. We particularly like the option to control it from a USB-connected laptop.

9/10

SaniaBOX

▶ Saniabox ▶ **saniabox.com** ▶ £144 / $169

A starter kit for teaching coding and electronics. **Rob Zwetsloot** checks out 13-year-old Sania's Kickstarter success story

SPECS

KIT CONTENTS: Raspberry Pi 4, preloaded 16GB microSD card, PSU and HDMI cables, keyboard, mouse, SaniaBOX add-on board, breadboard, LEDs, diodes

SENSORS: Thermal sensor, touch sensor, light sensor, optional gas sensor

RELEASE DATE: Sometime around or after April 2020

Despite how easy it is to put together a very specific physical computing teaching kit with a Raspberry Pi, there are actually not that many full kits that do so that include a mouse, keyboard, and Raspberry Pi.

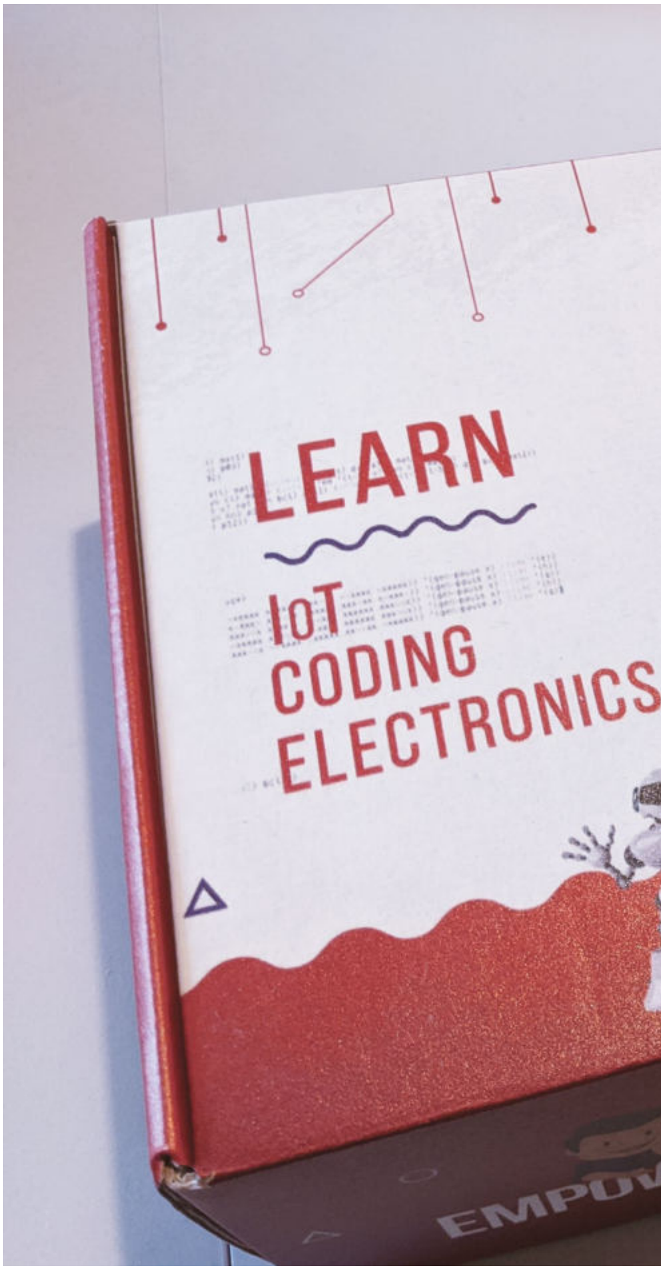
SaniaBOX is slightly different, though. Akin to a mix of the official Desktop Kit and specialised beginner's kits, the package comes with the requisite Raspberry Pi, keyboard, mouse, and microSD card along with the special SaniaBOX add-on board and a series of components.

This Kickstarter project brought to life was the idea of a Sania Jain, a 13-year-old who wanted to introduce coding to younger kids where possible. To that end, the all-important add-on board part of the kit includes a series of sensors, LEDs, as well as that big three-digit, seven-segment display.

Quick build

Unlike a pi-top or a Piper, you're not building a laptop or laptop-esque system – instead you're setting up a Raspberry Pi as normal and popping the HAT-like SaniaBOX add-on on top. Faster than even loading up your favourite streaming service (we'll catch up with you later, *Picard*) and it does allow you get stuck in straight away with some coding lessons.

▶ Sania is the girl behind the SaniaBOX



On the microSD card are a series of coding challenges, and you can find tutorials on the SaniaBOX website if you want to check out how the whole system works before diving in. The code for controlling the add-on bard can be simple (like with the LEDs) or a little more complicated (as with a 120-line script for working the seven-segment displays). The kit comes with some other LEDs, some diodes, and a breadboard so that you can do proper circuit prototyping once you graduate from some of the SaniaBOX add-on's functions.

The SaniaBOX add-on fits on Raspberry Pi much like a HAT

SaniaBOX comes in a great-looking box with all the components you need to start using and learning with Raspberry Pi

Simple and fun

The special add-on board works just fine, and has plenty of little sensors and ideas to keep younger folks – and even older folks new to making – entertained. With all the various functions, you can easily make something like a barometer – a great practical use of coding and electronics.

The price is possibly being reduced by the time you read this as well, so if the cost of it is putting you off a bit, it may well have about £15 / $20 knocked off. Considering that the Raspberry Pi Desktop kit with a very similar selection of components will set you back £100 / $120, we think it's a pretty good deal.

Verdict

A great and simple way to get younger makers (and even older newbies) into computing and coding.

8/10

10 Best: Audio projects

Want to hear something with a Raspberry Pi? Take note of these amazing audio projects

Raspberry Pi is not-so-secretly a multimedia powerhouse. With software like Kodi for TV streaming, RetroPie for a bit of gaming, and even jukebox software for your parties, it can handle just about anything. Here are some of our favourite ways that its audio skills have been put to use.

Wavepad

Wave away music

We like yelling at our digital home assistant thing to skip the track it's currently playing. It feels very Star Trek. Sometimes it's more dramatic to wave the music away – just like with the Wavepad.

magpi.cc/wavepad

Pi Zero Talking Radio

Pathé news, now!

This radio reads out your notifications from a variety of services. It doesn't have an old-fashioned news bulletin voice, but it's the spirit that counts.

magpi.cc/talkingradio

P.S.S.P. – Pi Single Song Player

One song only

Want to listen to one song, and one song only? Close a contact on this Raspberry Pi project for just that. Simple.

magpi.cc/pssp

Google Pi Intercom

Old-school personal assistant

Martin Mander made this with the AIY Projects kit that came with *The MagPi* #57. We love the meta idea of how this has been repurposed.

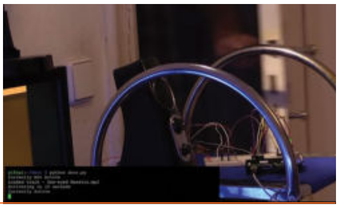
magpi.cc/intercom

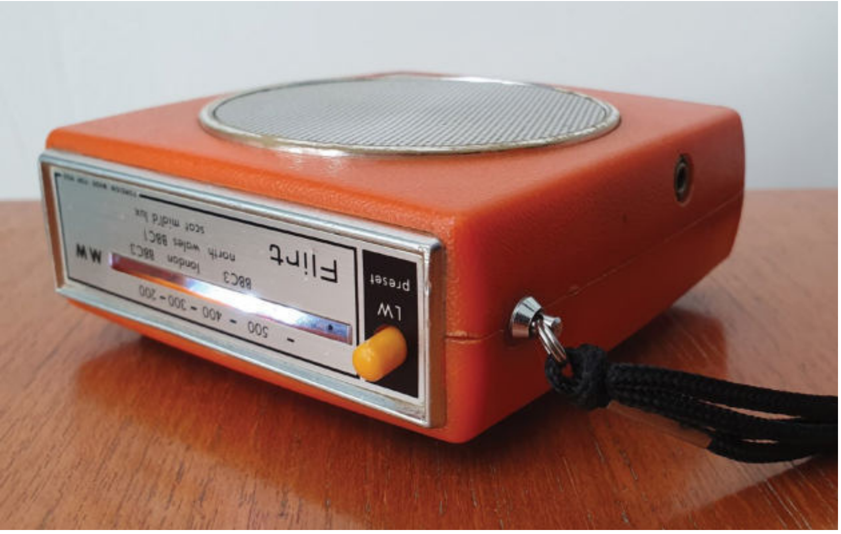
▸ Play your own theme tune

Dramatic music player

Feel like a sitcom character and have some slapping bass tunes play as you walk through the door. What's the deal with theme tunes, anyway?

magpi.cc/themetune





▴ Flirt Pi

Retro internet radio

This is a 1970 Flirt radio that upcycling maestro Martin Mander has turned into a Raspberry Pi-powered internet radio, without sacrificing much of its wonderful aesthetics.

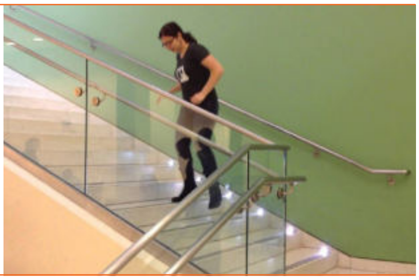
magpi.cc/flirtpi

▸ Piano Stairs

Tom Hanks inspired

Using a light tripwire to sense where you are, (carefully) dancing up and down these stairs should help with your scales and arpeggios.

magpi.cc/pianostairs

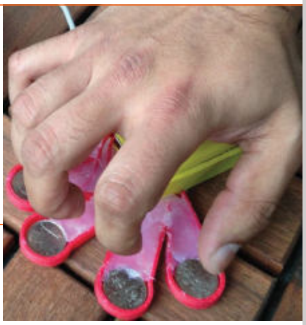


▸ Raspberry Pi music fingers

Finger drumming goodness

As well as being a lot of fun, this is a neat little conductivity experiment so you know how capacitive touch works. With a little beat added to it.

magpi.cc/musicfingers



▴ Ultrasonic theremin

Outer space vibes

This official Raspberry Pi project uses an ultrasonic distance sensor – something you mostly find on robots – to create a theremin sound as you move your hand through it.

magpi.cc/theremin

▾ Audio Radar

Accessible sodar

Using sound to detect distance is pretty standard tech, but it always helps to make it easier. This sodar project helps you do that.

magpi.cc/audioradar

Learn retro gaming with Raspberry Pi

Rediscover classic gaming and code your own 8-bit games with Raspberry Pi. By **Lucy Hattersley**

Code the Classics

CREATOR

Raspberry Pi Press

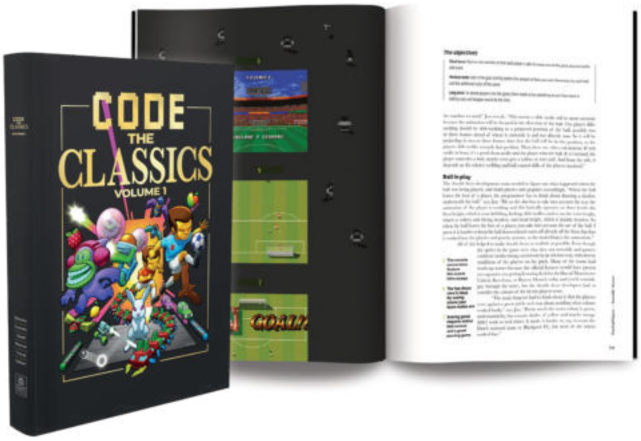
Price:
£12 / $15

magpi.cc/codetheclassics

If you're looking for the story behind some of the biggest games around, along with the details on how to program your own versions, look no further.

This gorgeous 224-page hardback book tells the stories of five seminal video games from the 1970s and 1980s. It starts in the style of Pong, then moves on to games similar to Bubble Bobble, Frogger, Centipede, and ends with a homage to Sensible Soccer.

All of the games are programmed using Python and Pygame Zero, with examples codeds by Raspberry Pi founder Eben Upton.



If you haven't got a copy yet, then you owe it to yourself to get one.

Retro game coding

Make sure you bookmark these websites

GettingStarted — wiki

Pygame Installation

Recent Releases

Further information if you need it.

Raspberry Pi

Windows installation

PYGAME

Most games you create on Raspberry Pi will involve Pygame. So make sure you bookmark the Pygame website. Here you'll find installation instructions and documentation.

pygame.org

REAL PYTHON

For a primer on Pygame, you should make sure to bookmark Real Python's website. It's a great overview of the basic concepts, up to sprites and collision.

magpi.cc/pygameprimer

PYGAME WIKI

There are far too many other resources for us to mention, but a great list is held on the Pygame wiki. There are tutorials here for every aspect of Pygame.

magpi.cc/pygamewiki

Retro Gaming with Raspberry Pi

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GAMING WITH RASPBERRY PI
164 PAGES OF VIDEO GAME PROJECTS
BUILD AN ARCADE MACHINE
BUILD AN ARCADE MACHINE
START

CREATOR

Raspberry Pi Press

Price:
£10 / $13

magpi.cc/retrogaming

While *Code The Classics* tells the story of retro games, and how to create your own versions, *Retro Gaming with Raspberry Pi* is all about building retro games consoles and arcade machines.

In it you'll discover how to emulate classic computers and consoles – and how to build a portable games machine, arcade cabinet, and even a pinball machine. You'll also discover how to upgrade classic computers, like a ZX Spectrum, with Raspberry Pi to bring the best of the new to the best of the old.

Make Games with Python

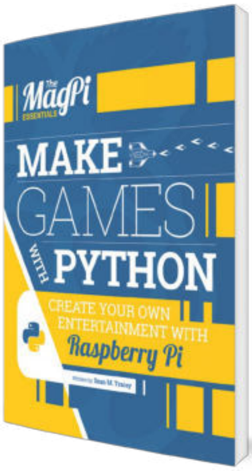
CREATOR

Sean M. Tracey

Price:
£3 / $3

magpi.cc/makegamespython

For a long time this Essentials guide was the final word on making Raspberry Pi with Pygame and Python games. Few books dive into the detail of Pygame as much as this. Sean M. Tracey takes you through the process of creating shapes and paths, movement, animation, and adding sound and music. It's a great book that culminates in building a shoot-'em-up complete with physics and forces. It's a bargain as well, currently on sale at just £2.50.


The MagPi ESSENTIALS
MAKE GAMES WITH PYTHON
CREATE YOUR OWN ENTERTAINMENT WITH Raspberry Pi
Written by Sean M. Tracey

Play the game

If you want to just play games, take a look at these sites

RETROPIE

RetroPie is the easiest way to turn a Raspberry Pi into a classic games console. Once you set this up, you'll be able to play all those classic games.
retropie.org.uk

PICO-8

PICO-8 turns Raspberry Pi into a fantasy console that's like an 8-bit gaming machine. You can download hundreds of games made by the community, and make your own games using the Lua language.
magpi.cc/pico8

ROMS

When making your own retro gaming console, you'll need to get ROMs (files) of classic games to play. Don't just go downloading ROMs off the internet, as you're likely to end up doing so illegally. Instead, take a look at our page of approved classic ROMs.
magpi.cc/legalroms

Dr Wednaud Ronelus

Educators can contact Dr Ronelus on **wednaudr@hotmail.com** for 'Socratic Discussion and other dialogues'

An educational researcher specialising in STEM, Raspberry Pi, and AI

- Day job **STEM Education Research Scientist**
- Community role **Educator**
- Website **magpi.cc/alchemistclub**

Hearing how some people refer to Raspberry Pi can be very humbling. However, when a STEM education researcher for New York City of over 20 years like Dr Wednaud Ronelus tells you that the release of Raspberry Pi meant nothing would be the same again, it's quite high praise.

"I have been testing cutting-edge ideas of how to bring a paradigm shift in STEM teaching and learning using the latest emerging technologies," Wednaud told us. "When the Raspberry Pi came out in 2012, I knew that education would never be the same again. I was totally blown away by this small credit-card-size computer."

Shortly after it came out, he got his own Raspberry Pi and started meeting with New York City's top technologists to learn more about it. He also attended early Raspberry Jams and events. One of the things to come out of this was Alchemist Club Studios.

▼ Students have access to some of the rare Google AIY kits

What is Alchemist Club Studios?
The Alchemist Club School & Studios is an educational platform that is designed to immerse learners in developing deep understanding of STEM concepts, ideas, and fundamental principles by using a 'physical approach' to computing in education. One of the main goals of the Alchemist Club School & Studios is addressing the crisis of minorities under-represented in STEM.

We have developed an innovative learning platform to help students develop a working understanding of bleeding-edge learning technologies… our first initiative focuses on using the Google Artificial Intelligence Yourself (AIY) Project Kits (Voice and Vision) to help learners develop a working understanding of machine learning, artificial intelligence, deep learning, and the Internet of Things by constructing learning artefacts that are used to assess STEM content mastery, critical thinking skills, and academic self-efficacy.

One of the main goals of the Alchemist Club School & Studios

▲ Students get to learn about a lot of AI and other computing subjects at Alchemist Club Studios

is to take advantage of the Google Classroom educational learning platform to develop a professional development portal for educators to learn how to construct the Google AIY projects and take it back to their classroom learning environment and enact the curriculum. We also focus on educational outreach using a global perspective. Since every educator has access to Google Classroom, we have opened the learning portal to educators all over the planet.

learning environments since 2013. I have collected tons of data and learning artefacts throughout the learning process.

Along the way, I have introduced many future electrical engineers, coders, and computer scientists to the magic of science via the Raspberry Pi.

What kind of teaching programs do you run?

I am running a cutting-edge STEM program in the Alchemist Club Studios to prepare the next generation of Google, Microsoft, Apple, Facebook, Amazon, IBM, Nvidia… etc. engineers. As Alan Kay once said, "The best way to predict the future is to invent it." That's exactly what we are doing in the Alchemist Club Studios. We have found an approach which can be used to address the 'digital divide' and the minorities under-represented crisis in STEM.

> " We've since trained over 1000 teachers through the programme "

Why Raspberry Pi?

The Raspberry Pi is a bridge to 'physical computing' in education. To be honest with you, it's the ideal platform that can be used to train the next generation of computer scientists, starting at elementary school level. I've been using the Raspberry Pi in my science

What kind of students do you get?

Well, that's a very difficult question; if you are talking about socioeconomic level, we are at the bottom of the ladder in the South Bronx, NYC. When it comes to health-related crises, the Bronx is the worst of the five boroughs in New York City. Let's not talk about poverty, housing, education, murder, suicide. You name it!

However, these social dilemmas have nothing to do with cognitive abilities. Many of my kids are currently at MIT and other prestigious colleges and universities all over the United States and abroad. Therefore, I know my instructional theory is measurable. I have found a way to galvanise them to stay focus in school because they love the cutting-edge projects we are doing in the learning space. Especially the Google AIY Project we have enacted. M

Switch of career

"My educational background is in the physical sciences. You might find this hard to believe: I left an MD/PhD program in Molecular Genetics-Gene Therapy to go into education because I understood the gravity of the minorities under-represented in STEM crisis. This reality hit me like a brick when I was taking some advanced courses in Physical Chemistry, Thermodynamics, Quantum Mechanics, Quantum Computing, and other emerging topics in the physical and biological sciences during my undergrad/graduate years. Most of the time, either I was the only African American student in the class or one of the very few who were astute enough to make it through the gauntlet. Quite honestly, this is a serious problem in education."

This Month in Raspberry Pi

#MonthOfMaking

You've been making and sharing – here's some of our faves!

At the time of writing, #MonthOfMaking is far from over. Many of you have been having a lot of fun creating and sharing already, though, so here's just a small selection of what we've seen so far!

01. We're not sure if semaphore is more or less functional than some train timetable apps

02. We've worked on its bigger sibling, PiGRRL 2, and while they're very well built, they can be tricky!

03. We always, always love seeing updates about this all-too-accurate Johnny 5 replica

04. Now that is a very cute screen

05. No it isn't

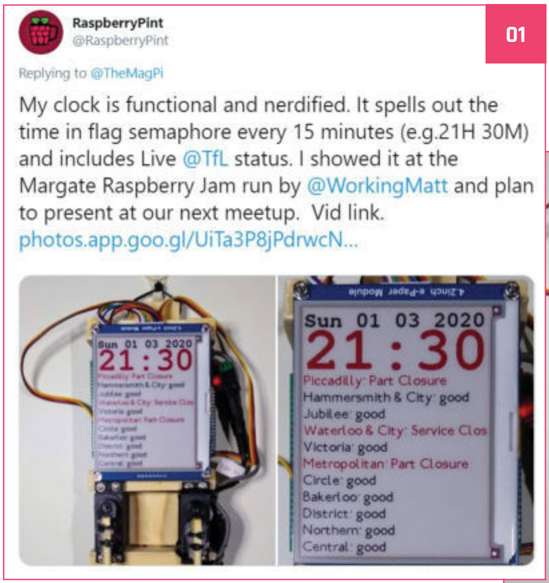
06. Our editor Lucy even got involved by checking out the new Adafruit 1.3" Bonnet!

07. Look out for more on this excellent project in a future issue!

08. Astronomy is a lot of fun and not many Raspberry Pi projects dabble in it

09. The Rusty Radio is a great looking, wood-enclosed internet radio box!

10. We've seen this project in action and it is simple, yet lovely!

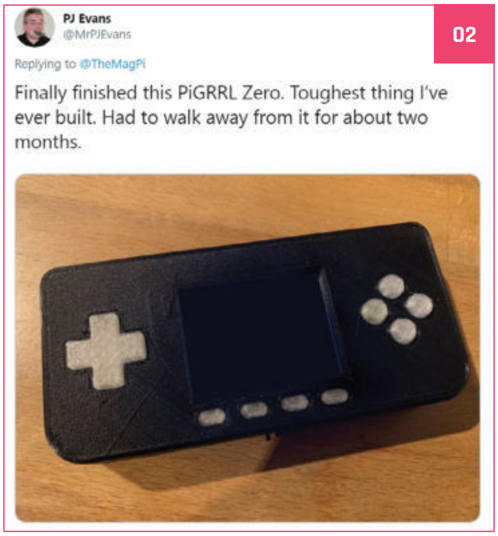


01

RaspberryPint
@RaspberryPint

Replying to @TheMagPi

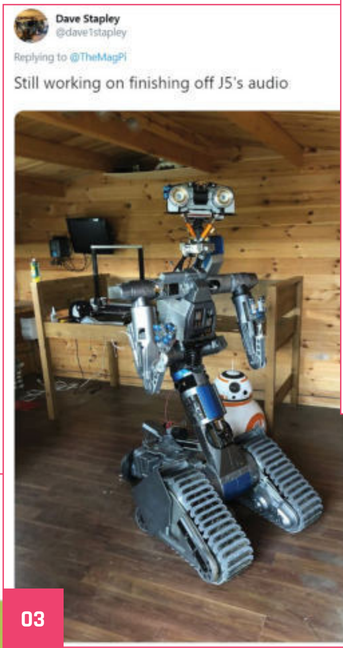
My clock is functional and nerdified. It spells out the time in flag semaphore every 15 minutes (e.g.21H 30M) and includes Live @TfL status. I showed it at the Margate Raspberry Jam run by @WorkingMatt and plan to present at our next meetup. Vid link. photos.app.goo.gl/UiTa3P8jPdrwcN...


02

PJ Evans
@MrPJEvans

Replying to @TheMagPi

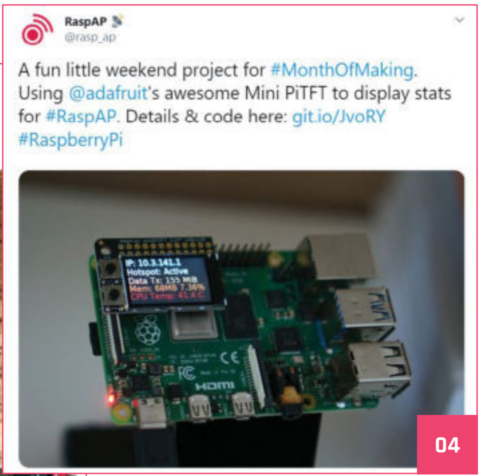
Finally finished this PiGRRL Zero. Toughest thing I've ever built. Had to walk away from it for about two months.


03

Dave Stapley
@dave1stapley

Replying to @TheMagPi

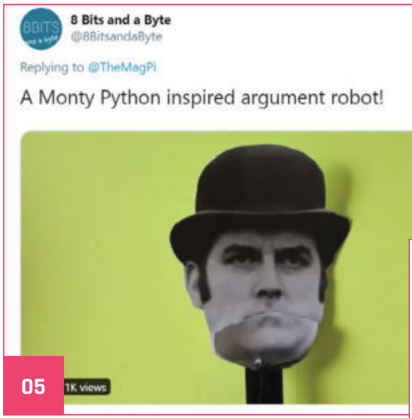
Still working on finishing off J5's audio


04

RaspAP
@rasp_ap

A fun little weekend project for #MonthOfMaking. Using @adafruit's awesome Mini PiTFT to display stats for #RaspAP. Details & code here: git.io/JvoRY #RaspberryPi

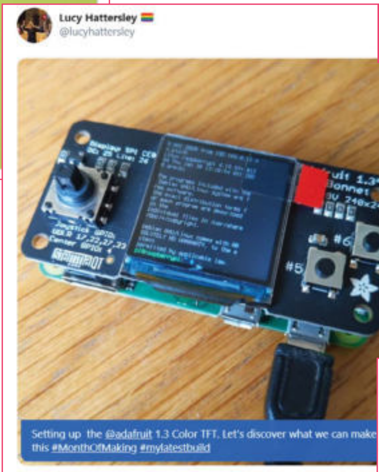
IP: 10.3.141.1
Hotspot: Active
Data Tx: 155 MiB
Mem: 68MB 7.36%
CPU Temp: 41.4 C


05

8 Bits and a Byte
@8BitsandaByte

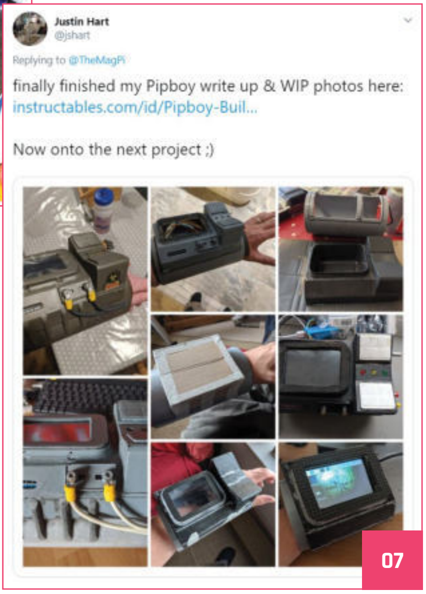
Replying to @TheMagPi

A Monty Python inspired argument robot!


06

Lucy Hattersley
@lucyhattersley

Setting up the @adafruit 1.3 Color TFT. Let's discover what we can make this #MonthOfMaking #mylatestbuild

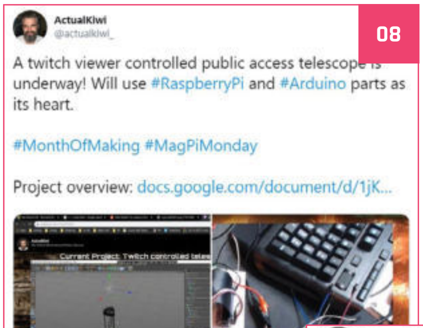

07

Justin Hart
@jshart

Replying to @TheMagPi

finally finished my Pipboy write up & WIP photos here: instructables.com/id/Pipboy-Buil...

Now onto the next project ;)

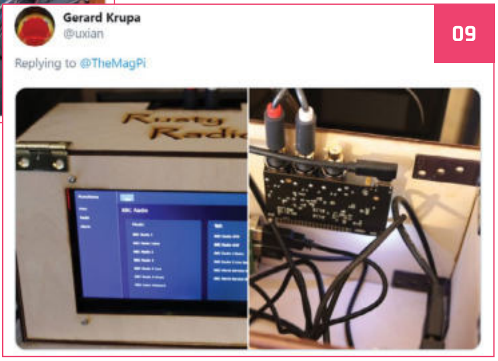

08

ActualKiwi
@actualkiwi_

A twitch viewer controlled public access telescope is underway! Will use #RaspberryPi and #Arduino parts as its heart.

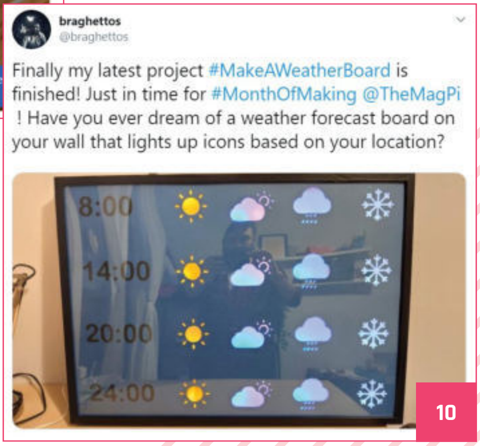
#MonthOfMaking #MagPiMonday

Project overview: docs.google.com/document/d/1jK...


09

Gerard Krupa
@uxian

Replying to @TheMagPi


10

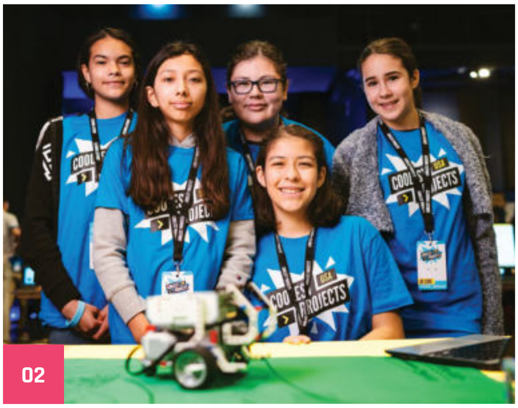
braghettos
@braghettos

Finally my latest project #MakeAWeatherBoard is finished! Just in time for #MonthOfMaking @TheMagPi ! Have you ever dream of a weather forecast board on your wall that lights up icons based on your location?

Coolest Projects USA 2020

Coolest Projects is back in LA to show off more amazing projects from younger makers!

A few weeks ago, Coolest Projects USA returned to the Discovery Cube to once again showcase incredible young makers. We weren't there this year, but it sounded as great as ever, and there are loads of amazing pics to prove it.


02

01

03

04

05

06

01. The venue is huge and bustles with maker activity!

02. There were many teams making many kinds of projects

03. Showing off sewing skills driven by making

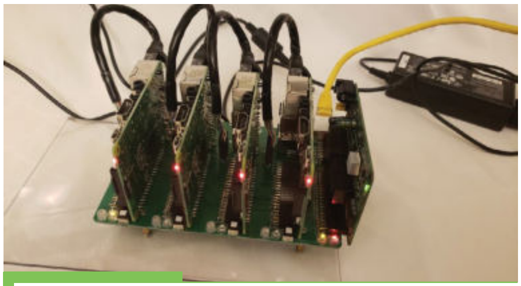
04. Accessibility projects, like this 'memory assistant', were also featured

05. Presenting your project is a big part of Coolest Projects

06. Winners are treated to a rain of ticker tape

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month



CloverPi

Looking for a way to create a Raspberry Pi-based cluster computer? CloverPi is one way you'll be able to achieve it using four Raspberry Pi boards. It's already hit its funding target, but there's still plenty of time to back it.

▶ **kck.st/2VwQn0d**

CROWDFUNDING A PROJECT?

If you've launched a Raspberry Pi-related project, let us know!

magpi@raspberrypi.org

Best of the rest!

Here are some other great things we saw this month

"APPLE WATCH IS PUNY"

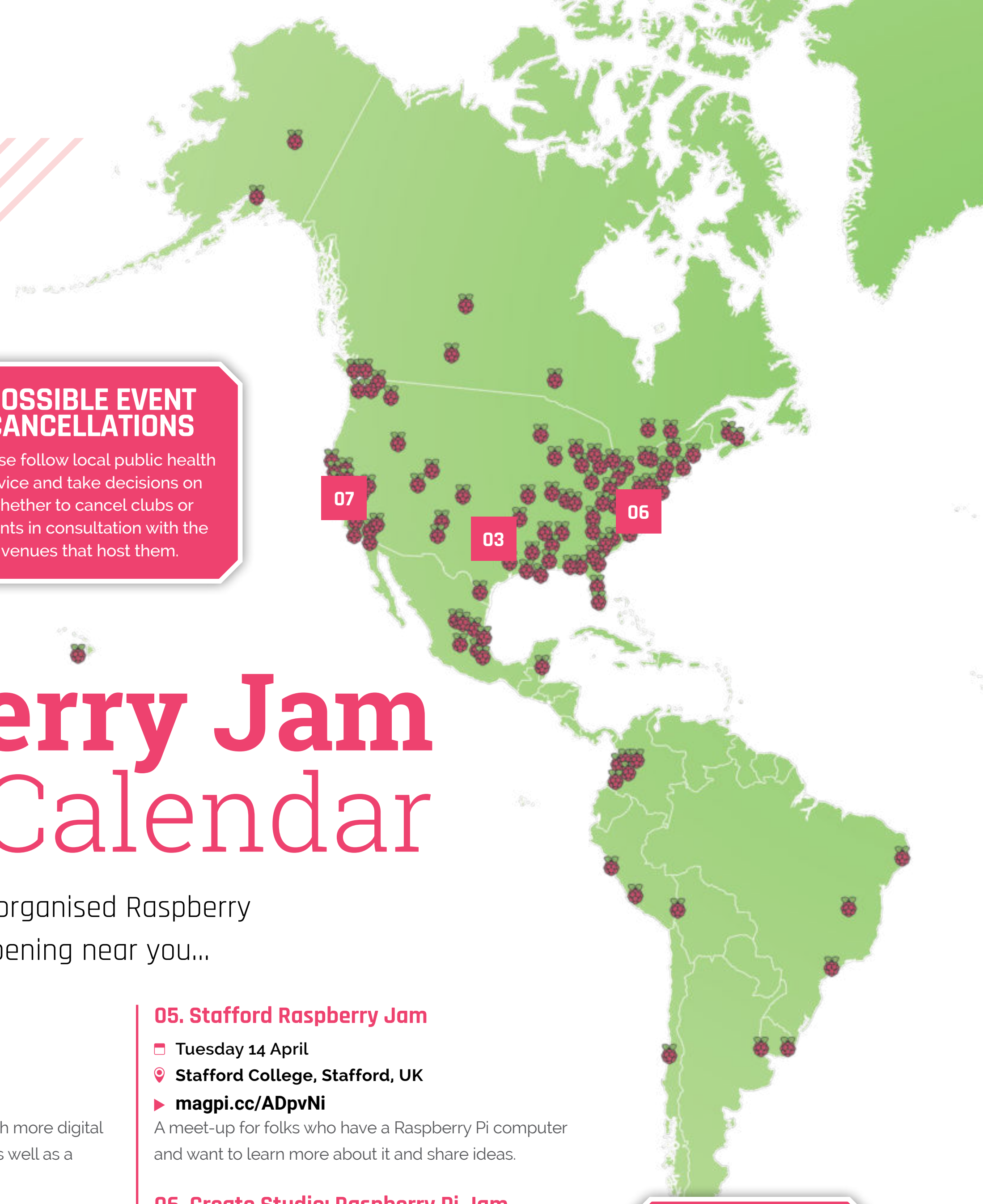
This is incredibly cyberpunk and we assume there are keyboards on their belt and giant power packs in a backpack.

▶ **magpi.cc/punywatch**

BATMAN 66 PINBALL

Hooking a pinball machine up to the internet is cool, but hooking a pinball machine up to the internet based on the 1966 Batman TV show is cooler.

▶ **magpi.cc/batman66**



Raspberry Jam Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

POSSIBLE EVENT CANCELLATIONS

Please follow local public health advice and take decisions on whether to cancel clubs or events in consultation with the venues that host them.

01. Leeds Raspberry Jam

Wednesday 1 April

Dixons Unity Academy, Leeds, UK

▶ **magpi.cc/ZMrFtM**

There will be chances to get hands-on with more digital making activities through the workshop, as well as a hackspace area.

02. Exeter Raspberry Jam

Saturday 4 April

Exeter Library, Exeter, UK

▶ **magpi.cc/BXyUyu**

A meeting for everyone interested in all things computers, microcontrollers, robotics, and making.

03. Dallas Young Makers Club

Saturday 11 April

J. Erik Jonsson Central Library, Dallas, TX, USA

▶ **dallasyoungmakers.org**

Free mentor-led hands-on projects for kids with Raspberry Pi, Lego Mindstorm robots, and more.

04. South Devon Tech Jam

Saturday 11 April

Paignton Library and Information Centre, Paignton, UK

▶ **magpi.cc/hgByVV**

A monthly informal and friendly session for anyone interested in technology, regardless of age or ability.

05. Stafford Raspberry Jam

Tuesday 14 April

Stafford College, Stafford, UK

▶ **magpi.cc/ADpvNi**

A meet-up for folks who have a Raspberry Pi computer and want to learn more about it and share ideas.

06. Create Studio: Raspberry Pi Jam

Sunday 19 April

Williamsburg Regional Library, Williamsburg, VA, USA

▶ **magpi.cc/iFHsjH**

Learn or practise a new skill. Each class will be a demonstration of a particular machine, with time for questions afterwards.

07. Castro Valley Jam

Saturday 25 April

Castro Valley Library, Castro Valley, CA, USA

▶ **magpi.cc/ipGYMb**

If you're interested in computer coding, you can tinker and code electronics at the Castro Valley Jam.

08. Raspberry Jam Zelzate

Saturday 25 April

Openbare Bibliotheek Zelzate, Zelzate, Belgium

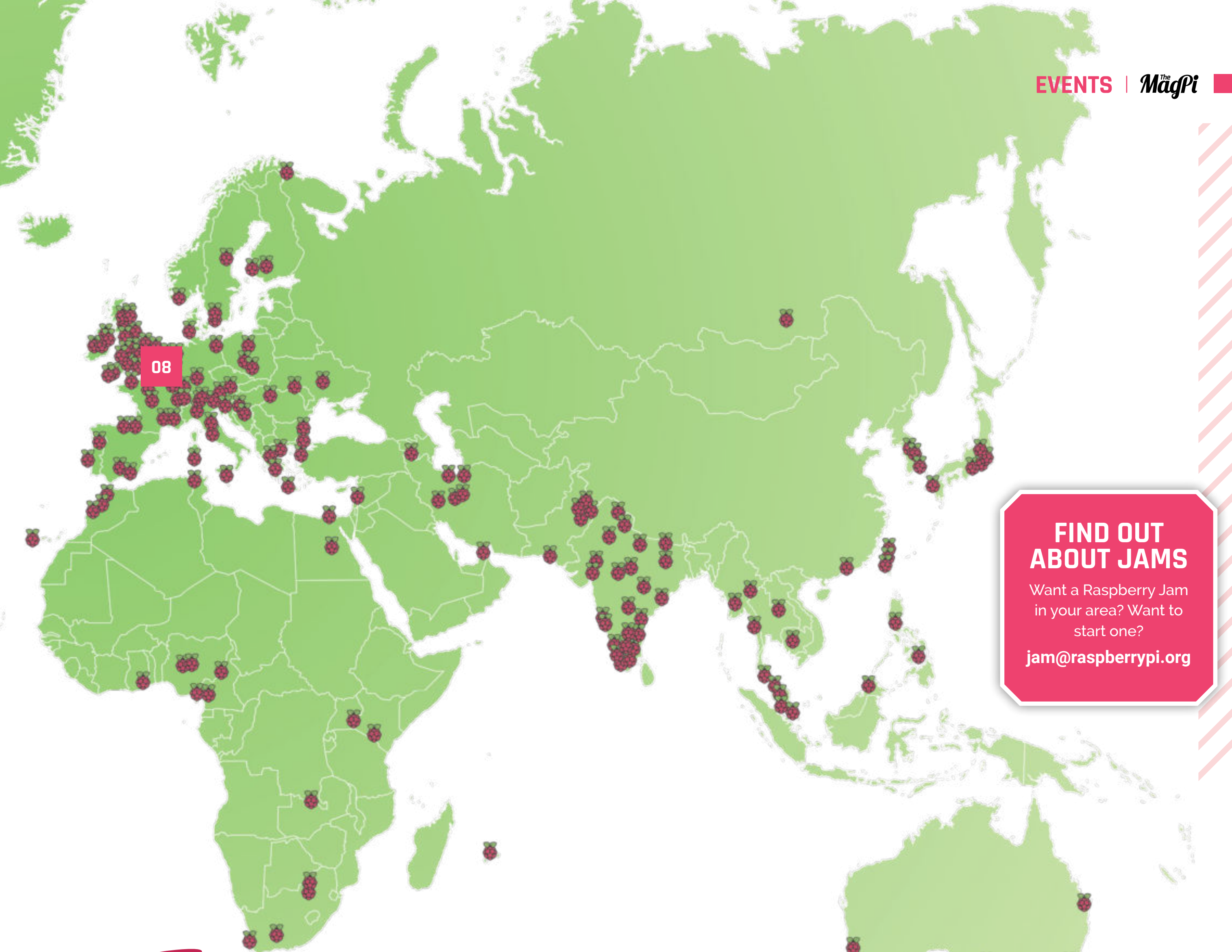
▶ **magpi.cc/UVVw5h**

Everyone is welcome to start, share, and work on their own project(s) in a fun and relaxed atmosphere.

FULL CALENDAR

Get a full list of upcoming events for April and beyond here:

rpf.io/jam

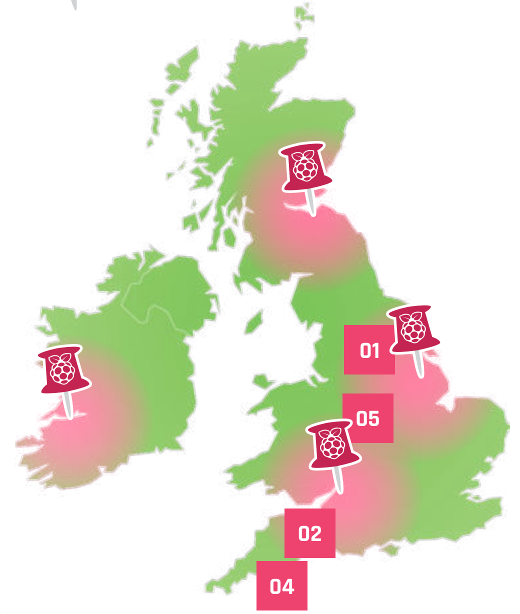


FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one?

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?



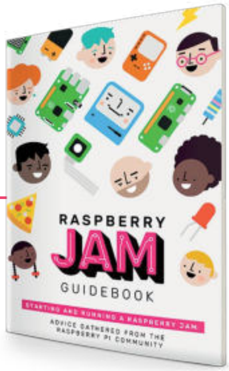
Raspberry Jam advice:

Risk assessment

"Risk assessment is about achieving a balance between a reasonable level of risk, and being able to get on with organising your activities. Remember, no activity is completely free from risk, and doing a risk assessment is not about making your activities risk-free."

Resource Centre

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **rpf.io/guidebook**


RASPBERRY
JAM
GUIDEBOOK
STARTING AND RUNNING A RASPBERRY JAM
ADVICE GATHERED FROM THE RASPBERRY PI COMMUNITY

Wireframe presents
Build Your Own
FIRST-PERSON SHOOTER
in Unity
Learn Unity • Create enemies • Design levels • Make Zombie Panic
Wireframe
BUILD YOUR OWN FIRST-PERSON SHOOTER IN UNITY
Level Design and Inspiration
The theory behind first-person hitboxes
Additional Mechanics
Creating a deployable gun turret in Unity

Your Letters

Reading at home

I'm looking for more stuff to read while at home and was wondering if there was a way to read all of *The MagPi* at all? Do you have like a big bundle magazine offer I can get from you at all?

Kath via Facebook

Yes and no – while we can't offer a big pack of all our magazines as we don't reprint, you can get every issue for free as a PDF from our website. Just head over to **magpi.cc/issues** to find them all.

The MagPi issue 90 out now!
February 2020
Free Raspberry Pi 4 Stand with the print edition of The MagPi magazine issue 90

Back Issues

Issue 89
January 2020
The 50 best Raspberry Pi tips & tools in The MagPi magazine issue 89
More details

Issue 88
December 2019
Rediscover retro computing and thermal-testing the new firmware. All in the latest edition of The MagPi magazine
More details

Remember, our PDFs are free from our website if you need another way to stock your library!

New release

```
import baby from mother&father
print("Hello World!")
```

Hi *The MagPi*,
My Daughter Zivah was born on Tuesday! Daddy's first present to her? A Raspberry Pi of course! (Although, I think it'll be a few years until it comes in use, so I might have to use it for now.)

Could I get this little photo attached and a small passage in *The MagPi* letters section, just as something so I can show her when she's older? #codersofthefuture!

Jeremy via email

Your wish is our command. We hope she gets to enjoy her Raspberry Pi in the future in whatever way she desires!

#MonthOfMaking the first

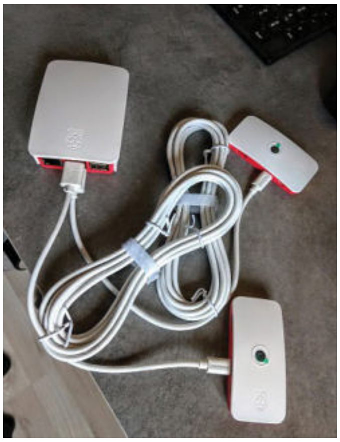
As [requested] in the last issue, please find in this email my project with Raspberry Pi I want to share: magpi.cc/babymonitor.

It's a dual baby monitor, made with one master Raspberry Pi 3 and two Raspberry Pi Zero with a Raspberry Pi Camera Module for each. Both Raspberry Pi Zero boards have no SD card, reducing the overall cost. Their firmware is fed by the master Raspberry Pi 3. I can watch both MJPEG streams on my local network and it works very well (at the time of writing, I have more than 150 days of uptime). It's robust, simple, and works with every web browser, even old ones.

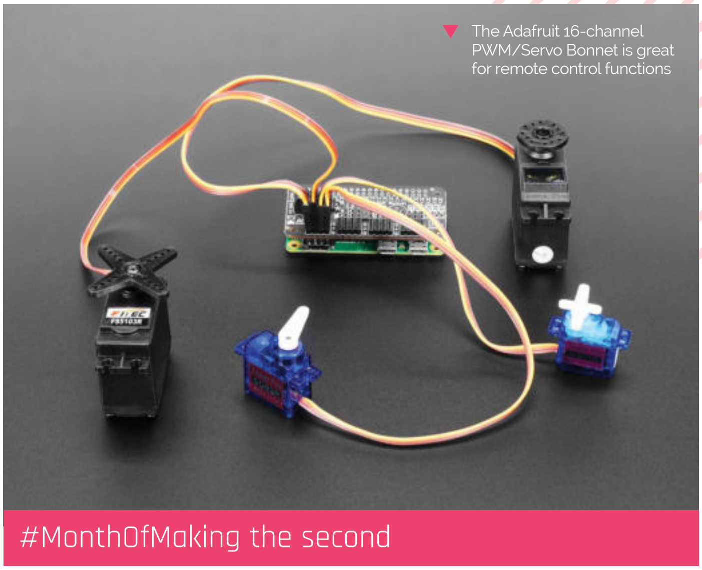
The tutorial is quite outdated and worth a good update, but some other forum members managed to achieve a functional setup.

Laurent via email

Wow, this is a cool project! We hope during the #MonthOfMaking you decide to give it a bit of an update for a more modern Raspberry Pi!



The use of white USB cables with the official case makes this very aesthetically pleasing



The Adafruit 16-channel PWM/Servo Bonnet is great for remote control functions

#MonthOfMaking the second

Please have a look at the tutorial to build a 'Remote control of model making with Raspberry Pi': magpi.cc/rcpi.

In the eighties, you had to build a lot yourself. This is different today. You only need a Phillips screwdriver for the right change and you can build the most beautiful model with prefabricated parts. The remote control is usually added almost for free.

Since I used to develop and build transmitters and receivers myself, I asked myself if this is possible with a Raspberry Pi.

Bernd via email

Model building and modifying is something we're quite fond of, so thank you for sending us this tutorial on how folks can do their own custom remote-controlling! If anyone uses it in their #MonthOfMaking projects, let us know!

Contact us!

Twitter	@TheMagPi
Facebook	magpi.cc/facebook
Email	magpi@raspberrypi.com
Online	raspberrypi.org/forums

Wireframe
£3
ALL FORMATS
LIFTING THE LID ON VIDEO GAMES
THE BEAUTIFUL GAME
The studios evolving the football genre
PLUS! Code your own Sensible Soccer homage
BLOODLINES 2
Inside Hardsuit Labs' toothsome sequel
ALTEREDVISTA
Games that recreate old user interfaces
TOOLBOX
Tips on world-building, narrative design, and more
DOUBLE STRIKE
YOUNG SOULS: THE BRAWLER WITH ATTITUDE TO SPARE
Issue 35 £3 | wfmag.cc
GIANT LEAP
Game jams as a route to industry success
STADIA AND BEYOND
Streaming: the future of video games?
INFIN
Epic space and ruthle
Ka-boom
When games look like comic books
On a mission
The secrets of great campaign design

WIN ONE OF TEN ARGON ONE CASES!

The Argon ONE case won our thermal performance test, and now you can win one!

> "The Argon ONE for Raspberry Pi 4 is the case that enhances the full potential of Raspberry Pi, from desktop computing, media and gaming to projects with GPIO."

Head here to enter: magpi.cc/win | **Learn more: magpi.cc/argonone**

Terms & Conditions

Competition opens on **25 March 2020** and closes on **30 April 2020**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

MAKE | BUILD | HACK | CREATE
132 PAGES EVERY MONTH
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
April 2020
Issue #29
FIGHTING ROBOTS
We test out a self-build kit
GAMES CONTROLLER
Get your games interacting with the real world
Men's Sheds
Charlie Bethel on tea, wellbeing and why your shed needs you
MAKE WITH WOOD
Pick the right tools
Master the materials
Build something useful
Arduino Displays
Testing NeoPixels
Recycling plant
PLAYING CARDS BUNNIES LITHOPHANES MUSIC

HOME AUTOMATION

MAKE YOUR SMART HOUSE WITH RASPBERRY PI DIY

THE MAGPI #93
ON SALE 30 APRIL

Plus!

The Octopede robot

Learn Java and electronics

Create your own magic mirror modules

ZX Spectrum Next Accelerated reviewed

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.com

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.com

Sub Editors
Phil King and Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Harriet Knight

Illustrator
Sam Alder

CONTRIBUTORS

Mike Cook, David Crookes, PJ Evans, Rosemary Hattersley, Nicola King, Simon Monk

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine.

ISSN: 2051-9982.

Come together, separately

Now is the time for us to stand together (a few feet apart). By **Lucy Hattersley**

I'll let you in on a secret: *The MagPi* magazine team has always been mobile. We work in various towns and cities, mostly in the UK.

There are Raspberry Pi Towers in Cambridge, and we return to the mothership from time to time, but we're happily floating around our parts of the world.

It's been a long time since I was a 9-to-5er chained to an office desk. When I first broke free from the office routine, I used the time and learned to program with MITx's Computational Thinking using Python XSeries (**magpi.cc/mitpython**).

> Print publications are wonderful, physical, tactile things

Coding is important, but I don't think it matters what you learn: paint a picture, learn cheese-making, or artificial intelligence computing paradigms, or how to play the flute; or make some incredible gadget with a Raspberry Pi and wires.

Learning from and teaching others is the 'root and heart and soul and centre' of Raspberry Pi. It's not just what we do: it's what we are.

Sharing for the soul

The other thing I love about Raspberry Pi is the community. I adore going to events like Pi Wars, and the Raspberry Jams, last year's Scratch Conference, and the birthday parties.

Community and sharing are more difficult right now, and I have little to add on that subject. In the words of the great Rod Stewart: "I don't want to talk about *it*."

Talking about *it* is a job for somebody more eloquent. Raspberry Pi Foundation's Chief Executive Philip Colligan has written a wonderful blog post about the organisation's response (**magpi.cc/eventresponse**)

From the many wise words in Philip's post, this caught my eye: "[Raspberry Pi] has always been an organisation and a community where people genuinely care about and support each other. Let's all double down on that now."

The MagPi magazine is our refuge. Here is where we all come to learn about the amazing projects being built by the community; it's a platform for you to share your creations.

We are a sociable bunch. You'll find us on Facebook (**magpi.cc/facebook**) and Twitter (**magpi.cc/twitter**). Be especially sure to join in on our #MyLatestBuild and #MagPiMonday Twitter chats. If you don't care for social media, we post stories daily on our website (**magpi.cc**). Feel free to add comments below each story. Write to us at **magpi@raspberrypi.com**; we will respond.

And please take out a subscription to a magazine. I don't mind which one you subscribe to (obviously, there are one or two I recommend).

Print publications are wonderful, physical, tactile things. They are made with love by people who care; curated by experts and written by heartfelt enthusiasts. They pop through the door once a month like a surprise present, and they always cheer me up.

Magazines – even if they think they're about caravans, or fishing, or knitting – are always about community. The word comes from the French 'magasin' meaning, 'storehouse'; it's about collecting things together: in this case, people.

There's a nerd joke: 'Now is the time for us to come together; separately; in our own homes.' Drum rolls; everybody laughs.

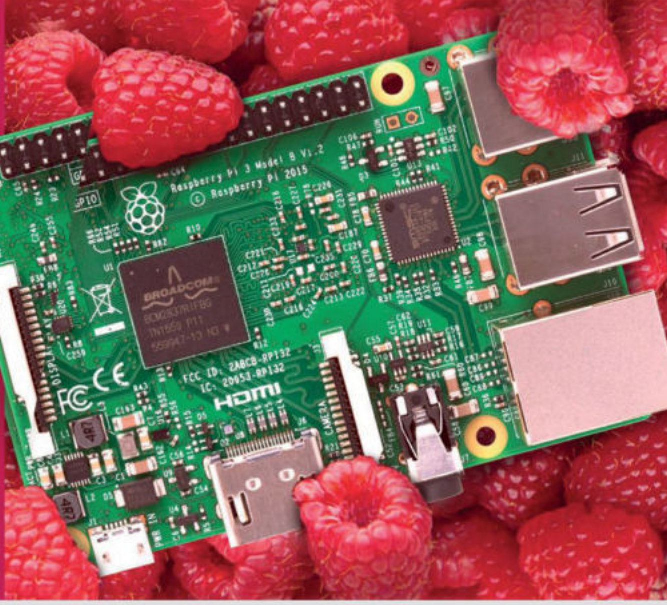
There are lots of ways we can come together. Let's use them.

AUTHOR

Lucy Hattersley

Lucy is editor of *The MagPi* magazine. Social-distancing amateur-professional. May be distracted by Animal Crossing. Bit nerdy but really likes to hug, which is awks at the best of times.

magpi.cc



PIPER
Computer Kit
Mission Zero:
The blueprint to your Piper Computer
CLASSIFIED